Development of a Miniature, Long-Duration GPS Tag

Grant MacLean MEng CEng MIET

Thesis submitted for qualification of

PhD in Electronics

Heriot-Watt University

School of Engineering & Physical Sciences

January 09

Abstract

This research concentrates on both weak GPS signal-processing and GPS time error correction in parallel with the development of an innovative GPS tag – "TrackTag". The technology is aimed at animal tagging applications and therefore has particular challenges. The tag has to be small enough so as not to impact the behaviour of the animal under study as well as robust enough to operate wherever the animal goes. TrackTag's architecture differs form that of a conventional GPS receiver in that it uses very short snapshots and there is no processing performed on the tag itself. This enables the tag to operate for over a year on a small battery.

Weak signal processing algorithms studied include FFT techniques and conventional correlators for demodulation of the GPS signals using both non-coherent and coherent integration. The techniques developed enabled the tracking of signals down to 24dB-Hz, which is comparable to today's best conventional GPS units. However, the algorithms developed during this research achieve such performance using only 76 milliseconds of data while conventional designs require many seconds of data. The performance of the navigation solutions was not compromised in terms of accuracy. A 2dRMS accuracy of 29 metres was observed under forest canopy and the research even demonstrated over 50% success rate in the Amazon Rainforest canopy where conventional GPS tags had never been shown to work.

New time correction techniques were also required as it was estimated that a 100ms clock error would result in 100m positional error. Temperature calibration of the RTC crystal was developed as the first step. However, with that, the clock drift over a month could still be up to 5 seconds (2ppm) and so it became clear that for long-duration (>3 months) studies another method for maintaining accuracy was required. Over-determined navigation solutions, and their subsequent position residual error, were used to estimate the time offset with some success. Pattern-matching against WAAS signals offered exceptional performance and was demonstrated on real-world tracking studies that were running concurrently with this research. The accuracy was considered to be limited only by the front-end sample period, i.e. 200 nanoseconds for any given deployment duration.

Acknowledgements

Peter Brown, NAVSYS Ltd

As Managing Director of Navsys Ltd, Peter ran the commercial business which owns both the existing TrackTag technology and any other intellectual property created by this research. Peter was also responsible for design work on the TrackTag hardware. He led the design of firmware for the tag and Interrogator units as well as all TrackTag packaging development.

Dr Paul Record, Heriot-Watt University

Acting as academic supervisor, Paul guided the progress of the research with a focus on achieving the academic goals necessary for the award of PhD.

Dr Charles Bishop, Bangor University

Charles was a primary customer for the TrackTag product during this research. His involvement opened doors to suitable biologists undertaking research that would benefit from the use of TrackTag.

Mathias Tobler, University of Austin, Texas

Mathias was a primary user of TrackTag during the research. He was based in the Peruvian Amazon conducting research involving the tracking of Tapir.

Table of Contents

List of Tables……x

List of Figures……xi

List of Abbreviations xvi

Publications……..xix

Presentations……..xix

Prizes……..xix

Chapter 1: Introduction 1

1.1 Thesis Motivation 1

1.2 History of Animal Tracking Telemetry.....3

1.3 Animal Tracking Requirements 8

1.3.1 Tag Weight 8

1.3.2 Tag Packaging 10

1.3.3 Data Retrieval 11

1.3.4 Accuracy.....13

1.4 Animal Tracking with Global Navigation Technologies 15

1.4.1 OMEGA Background 15

1.4.2 OMEGA in Animal Tracking.....17

1.4.3 TRANSIT 17

1.4.4 Argos Background.....19

1.4.5 Argos in Animal Tracking 20

1.4.6 GPS Background.....21

1.4.7 GPS in Animal Tracking 23

1.5 Summary of Current Animal-Tracking Technologies 26

Chapter 2: GPS in Animal Tracking 28

2.1 Key Factors in GPS Design 28

2.1.1 Antenna design 28

2.1.2 Signal Detection 29

2.1.3 Receiver Sensitivity 34

2.1.4 Almanac and Ephemeris 38

2.1.5 Navigation Solution *38*
2.1.6 Time-To-First-Fix *40*
2.1.7 Power *41*
2.2 GPS Augmentations 42
2.2.1 Differential GPS *42*
2.2.2 Assisted GPS *42*
2.2.3 WAAS *43*
2.2.4 Snapshot Processing *44*
2.2.5 Fastloc™ *44*
2.2.6 TIDGET™ Technology *45*
2.3 TIDGET in Animal Tracking (TrackTag) 50
2.3.1 Performance *50*
2.3.2 Duration *50*
2.3.3 Size *51*
2.3.4 Market Positioning *52*
2.3.5 Early Results *53*
2.4 Focus for TrackTag Research 55

Chapter 3: Antenna Integration 57

3.1 Overview 57
3.2 Turntable Concept 59
3.3 Turntable Results 64
3.4 Summary 68

Chapter 4: Signal Detection 69

4.1 Tracking Correlator Technique 70
4.1.1 1ms Correlator Engine *70*
4.1.2 Tracking Loop Design *72*
4.1.3 Real Signal Detection *75*
4.1.4 Signal Power *79*
4.1.5 Correlator Accuracy *83*
4.2 FFT Technique 85
4.2.1 Overview *85*
4.2.2 Real-World Signal Detection *87*

4.2.3 FFT Quantization *89*
4.2.4 FFT Accuracy *91*
4.3 Comparison of Techniques 94
Chapter 5: Weak Signal Detection 95
5.1 Introduction 95
5.2 Weak GPS Signal Detection Review 96
5.2.1 Signal Integration *96*
5.2.2 Block Accumulation *97*
5.2.3 Non-Coherent Data Bit Inversion Mitigation *97*
5.2.4 Multiple Data Bit Coherent Integration *98*
5.2.5 Cross-Correlation *100*
5.3 Non-Coherent Integration 103
5.3.1 Carrier-to-Noise (C/N_0) Calculation *104*
5.3.2 Thresholding *106*
5.4 Coherent Integration 107
5.4.1 All-Bit-Permutation Accumulation Technique *109*
5.4.2 Coherent Carrier-to-Noise (C/N_0) Calculation *113*
5.4.3 Threshold Setting *115*
5.5 Peak Detection 117
5.5.1 C/N_0 Thresholding *117*
5.5.2 Peak Noise *117*
5.5.3 Cross-correlation Impact on Thresholding *118*
5.6 Coherent & Non-Coherent Integration Summary 120
5.6.1 Noise Floor *120*
5.6.2 Bandwidth *122*
5.7 Static Point Tests in Amazon Rainforest 124
5.8 Extended Snapshot Integration 127
5.8.1 Extended Integration Algorithm Development *127*
5.8.2 Extended Integration Trials *130*
5.8.3 Summary *133*
Chapter 6: Signal Detection Enhancements 134
6.1 Doppler Search Reduction 134

6.1.1 Receiver Velocity *134*
6.1.2 Oscillator drift *135*
6.1.3 Time Error *136*
6.1.4 Position Error *138*
6.2 WAAS Navigation 143
6.2.1 Background *143*
6.2.2 Implementation *144*
6.2.3 Approximating Position using Polynomials *146*
6.2.4 Interpolation Summary *150*
6.2.5 WAAS Navigation Results *150*
Chapter 7: Time Correction 152
7.1 Motivation 152
7.2 Accuracy Measurement Overview 153
7.3 Temperature Correction 157
7.4 Endpoint Adjustment 161
7.5 Reduction of residuals 162
7.5.1 Background *162*
7.5.2 Hypothesis *163*
7.5.3 Algorithm Development *163*
7.6 Residual Reduction Technique (RRT) and Temperature Correction Results 168
7.6.1 Raw Comparison *168*
7.6.2 Temperature Correction and RRT combination *169*
7.6.3 Filtering using Points of High Confidence *171*
7.6.4 Results from using RRT Adjustment *173*
7.7 WAAS Signal Pattern matching 175
7.7.1 Hypothesis *175*
7.7.2 Statistical Analysis of WAAS symbol patterns *175*
7.7.3 Algorithm Description *177*
7.7.4 Results from WAAS Time Correction *179*
Chapter 8: Results & Conclusion 183
8.1 British Antarctic Survey (BAS) Results 183

8.2 University of Bangor Results 186
8.3 Moore Foundation / University of Austin Results 187
8.4 Ongoing Research using TrackTag 188
8.5 Conclusion 189

Appendix A: Position Estimation 190

A.1 Pseudo-range 190
A.2 The Geometry Matrix 191
A.3 Dilution of Precision 192

Appendix B: Platform Development 193

B.1 Interrogator Development 195
B.1.1 Software Development 195
B.1.2 Low-Level Control 203
B.1.3 Summary 204
B.2 Tag Design 206
B.2.1 Front-End Description 207
B.3 Algorithm Development 208
B.3.1 Signal Detection 209
B.3.2 Position Estimation Algorithm 211

Appendix C: SBAS Broadcast Data File Description 213

References 215

List of Tables

Table 1: Animal-Tracking Technology Comparison 27
Table 2: Wildlife Tracking Market Split by Technology 53
Table 3: SIGN & MAG bit to value 80

List of Figures

Figure 1. 2004 VHF Radio-Telemetry Market Spilt 6
Figure 2. BPSK Spreading 30
Figure 3. Gold Code Generator 31
Figure 4. Basic Bit Correlation 33
Figure 5. Conventional GPS Receiver Block Diagram 34
Figure 6. Receiver Architectures 48
Figure 7. Comparison of Energy Per Fix 50
Figure 8. Comparison of Maximum Number of Fixes 51
Figure 9. Flight of Albatross in South Atlantic 54
Figure 10. Turntable Hardware Setup showing Tag to Satellite angles (Az_{ANT}, Az_{SAT} and El_{SAT}) 60
Figure 11. Satellite Elevation Prediction 61
Figure 12. Typical Turntable Measurement Points 62
Figure 13. 'Azel' Gain Plot for Whip Antenna 64
Figure 14. Azimuth Gain Plot for Whip Antenna 65
Figure 15. Elevation Gain Plot for Whip Antenna 65
Figure 16. 'Azel' Gain Plot for Ceramic Chip Antenna 66
Figure 17. Azimuth Gain Plot for Ceramic Chip Antenna 66
Figure 18. Elevation Gain Plot for Ceramic Chip Antenna 67
Figure 19. 1ms Correlator Engine 71
Figure 20. Tracking Correlator Design 72
Figure 21. Correlator Tracking Code-Phase 75
Figure 22. Tracking Correlator Power versus Code Phase Offset 76
Figure 23. Correlator Tracking Doppler 77
Figure 24. Tracking Correlator Power versus Doppler Offset 78
Figure 25. Correlator Frequency Measurement over swept initial Doppler estimate 78
Figure 26. Tracking Correlator Power Surface 79
Figure 27. Percentage of time MAG bit set using Passive Antenna 80
Figure 28. Percentage of time MAG bit set using Active Antenna 80
Figure 29. Mean Sample Value of Raw Data using Active Antenna 81
Figure 30. Measured Millisecond Noise Floor (squared) 82

Figure 31. Measured Millisecond Noise Floor (Non-squared) 82
Figure 32. Estimated Frequency Error Using Correlator 83
Figure 33. Estimated Code-Phase Error Using Correlator 84
Figure 34. Basic FFT Process 86
Figure 35. 1 ms FFT Output 87
Figure 36. Typical FFT Output over Doppler 88
Figure 37. Typical FFT Output over Doppler and milliseconds 88
Figure 38. Linear Sub-Sample Adjustment 89
Figure 39. Residual Error with Linear Sub-Sample Correction 90
Figure 40. Horizontal Error with/without Linear Sub-Sample Correction 91
Figure 41. Estimated Code-Phase Error Using FFT 92
Figure 42. Estimated Code-Phase Error Using FFT with Sub-Sample Adjustment 93
Figure 43. Estimated Frequency Error Using FFT with Sub-Doppler Bin Adjustment 93
Figure 44. Cross-Correlation from PRN-22 (Satellite provide 51dB-Hz Signal) on PRN-1 channel (Satellite not in view) 101
Figure 45. Non-Coherent Integration 103
Figure 46. Mean Non-Coherent Convolution Value 104
Figure 47. Non-Coherent Normalised Accumulation Mean and Std Deviation 105
Figure 48. Non-Coherent 99% PFA C/N0 106
Figure 49. Non-Coherent PFA C/N0 107
Figure 50. Coherent Integration 107
Figure 51. Possible Bit-Snapshot Alignment 110
Figure 52. Partial Accumulations 111
Figure 53. Coherent Accumulation Mean and Expected Signal Level 114
Figure 54. Coherent Normalised Accumulation Mean and Expected Signal Level 114
Figure 55. Coherent 99% PFA C/N0 115
Figure 56. Coherent PFA C/N0 116
Figure 57. Noise Floor with/without Strong Signal Present (PRN-12 did not exist at that time) 118
Figure 58. Comparison between Coherent & Non-Coherent Noise Floor 121
Figure 59. Comparison between Coherent & Non-Coherent Bandwidth 122
Figure 60. Coherent & Non-Coherent Bandwidth using real signal 123
Figure 61. Amazon Static Position Points - Minimum C/N0 Signals Detected 124

Figure 62. Amazon Static Position Points - Average No. of Signals Detected 125
Figure 63. Amazon Static Position Points - Fix Success Percentage..... 125
Figure 64. Amazon Static Position Points - 2dRMS Error 126
Figure 65. Raw and Filtered Phase (Measurement acquired by taking ratio of I & Q correlator outputs) 127
Figure 66. Residual Phase Drift after Inverse Phase Correction applied 128
Figure 67. Corrected Phase on Raw Data 129
Figure 68. Generated Phase LUT for 25 Snapshots 129
Figure 69. Derivative of Generated Phase LUT for 25 Snapshots..... 130
Figure 70. Forest Data - Signal Strength..... 131
Figure 71. Forest Data - No. Of Signals Detected 132
Figure 72. Forest Data - Positions..... 132
Figure 73. Forest Data - Position Error..... 133
Figure 74. Oscillator Temperature Curve for 00208925..... 135
Figure 75. Calibration Doppler Error..... 136
Figure 76. Error in Doppler prediction due to time offset 137
Figure 77. Error in Doppler prediction with satellite acceleration..... 138
Figure 78. Error in Doppler prediction due to position error..... 139
Figure 79. Azimuth Angle and Doppler Error surface direction 139
Figure 80. Line-Of-Sight Velocity direction and Doppler Error surface direction..... 140
Figure 81. Doppler Error with Position Error in line-of-sight velocity direction 141
Figure 82. Predicted Versus Measured Maximum Doppler Error 141
Figure 83. Photo of Geostationary Satellite tracks..... 143
Figure 84. WAAS 3D position over 11 days 145
Figure 85. WAAS X, Y and Z position over 11 days 146
Figure 86. Polynomial Approximation of Z position over 11 days 147
Figure 87. Polynomial over 1 day..... 147
Figure 88. Polynomial Error using Neighbouring day points..... 148
Figure 89. Z Error using hourly polynomials 149
Figure 90. Anomoly in X position data..... 149
Figure 91. "Romina" without use of additional WAAS signal 150
Figure 92. "Romina" with use of additional WAAS signal 151
Figure 93. Position Error Growth with Increasing Time Offset 154

Figure 94. Drift over Time Offset 155
Figure 95. Accuracy over Time Offset 155
Figure 96. Tag Calibration Temperature and Fractional Seconds 158
Figure 97. Time Drift against Temperature in Calibration 159
Figure 98. Distribution of RTC Drift after Temperature Calibration 160
Figure 99. Satellite Count 164
Figure 100. Residual Error over time for snapshot 39 165
Figure 101. Residual Error over time for snapshot 3019 166
Figure 102. Residual Error over time for snapshot 672 167
Figure 103. Residuals Reduction Time Estimation Versus Temperature Correction 168
Figure 104. Difference between RRT and Temperature Correction 170
Figure 105. Raw and Filtered RRT using Points of High Confidence 171
Figure 106. Temperature Correction with/without Filtered RRT Adjustment 172
Figure 107. East Movement without RRT Adjustment 173
Figure 108. East Movement with RRT Adjustment 173
Figure 109. East Movement without RRT Adjustment (zoomed in to end of dataset) 174
Figure 110. East Movement with RRT Adjustment (zoomed in to end of dataset) 174
Figure 111. 12-bit Pattern Histogram for daily broadcast WAAS symbols 176
Figure 112. Non-Phase-Ambiguous 12-bit Pattern Histogram for daily broadcast WAAS symbols 177
Figure 113. WAAS Correction Points for "Romina" Dataset 179
Figure 114. WAAS Correction Points (zoomed) 180
Figure 115. WAAS Correction "Noise" 180
Figure 116. WAAS Correction "Noise" with Code-Phase Adjustment 181
Figure 117. WAAS Correction Points with Code-Phase Adjustment 182
Figure 118. "Romina" Dataset East Movement 182
Figure 119. Albatross Track in South Atlantic 184
Figure 120. King Penguin Track in South Atlantic 185
Figure 121. Leatherhead Turtle Track in Zante 186
Figure 122. Jaguar Track in Amazon Rainforest 187
Figure 123. TrackTag System 194
Figure 124. Software Top-Level Design 196
Figure 125. Main Window 197

Figure 126. Initialise Window 198
Figure 127. Monitor Window 199
Figure 128. Download Window 200
Figure 129. Interrogator Info Screen 201
Figure 130. Tag Info Screen 202
Figure 131. Battery Management Window 203
Figure 132. Flowchart for usual TrackTag operation 204
Figure 133. Tag Photo (Top Side) 206
Figure 134. Tag Photo (Bottom Side) 206
Figure 135. Front-End Block Diagram 207
Figure 136. Front-End Frequency Plan 207
Figure 137. Gold Code Spreading 209
Figure 138. Navigation Solution Top Level 212

List of Abbreviations

1dRMS	1-Deviation Root-Mean-Square (RMS)
2dRMS	2-Deviation Root-Mean-Square
3D	3 dimensional
3dRMS	3-Deviation Root-Mean-Square
AGC	Automatic Gain control
APL	Advanced Physics Laboratory
BPSK	Binary Phase Shift Keying
C/N_0	Carrier to Noise Ratio
CA	Coarse/Acquisition
CEP	Circular Error Probable
CNES	Centre National d'Études Spatiales
dB	Decibel
dB-Hz	Decibels per 1Hz bandwidth
dBm	Decibels referenced to 1 milliwatt
dBW	Decibels referenced to 1 Watt
DC	Direct Current
DGPS	Differential Global Positioning System
DLL	Delay Locked Loop
DOP	Dilution Of Precision
DSP	Digital Signal Processing
DUT	Device Under Test
EEG	Electroencephalography
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FFT	Fast Fourier Transform
FLL	Frequency Locked Loop
GB	Giga Byte
GHz	Giga Hertz
GPS	Global Positioning System
GSM	Global System for Mobile Communications

GUI	Graphical User Interface
HDOP	Horizontal Dilution Of Precision
Hz	Hertz
IF	Intermediate Frequency
kHz	Kilohertz
km	Kilometre
LEO	Low Earth Orbit
LNA	Low Noise Amplifier
LOP	Line Of Precision
LOS	Line Of Sight
m	Metre
MAG	Magnitude
MHz	Megahertz
mm	Millimetre
ms	Millisecond
nm	Nautical Mile
NMEA	National Marine Electronics Association
NOAA	National Oceanic and Atmospheric Administration
OEM	Original Equipment Manufacturer
PC	Personal Computer
PDOP	Position (3D) Dilution Of Precision
PFA	Probability of False Alarm
PLL	Phase Locked Loop
pps	Pulse-Per-Second
PRN	Pseudo-Random Noise number
PTT	Platform Terminal Transmitters
RF	Radio Frequency
RHCP	Right-Hand Circularly Polarised
RMS	Root-Mean Square
RRT	Residuals Reduction Technique
RTC	Real-Time Clock
s	Seconds
SBAS	Satellite Based Augmentation System

SMRU	Sea Mammal Research Unit
SMS	Short Message Service
SNR	Signal to Noise Ratio
SV	Space Vehicle
TDOA	Time Difference Of Arrival
TTFF	Time To First Fix
U.S.	United States
UHF	Ultra High Frequency
UTC	Coordinated Universal Time
V	Volts
VDOP	Vertical Dilution Of Precision
VHF	Very High Frequency
VLF	Very Low Frequency
WAAS	Wide Area Augmentation System

Publications

1. "Virtual Anechoic Chamber using GPS Signals", Grant MacLean. The Journal of Navigation, Volume 60, No. 1, Jan 2007.

2. "Novel GPS tracking of sea turtles as a tool for conservation management", G. Schofield, C. Bishop, G. MacLean, P. Brown, M. Baker, K. A. Katselidis, P. Dimopoulos, J. D. Pantis & G. C. Hays. The Journal of Experimental Marine Biology and Ecology, Mar 2007.

3. "Weak GPS Signal Detection in Animal Tracking", Grant MacLean. The Journal of Navigation, Volume 62 No1, Jan 2009.

NOTE: Further publications of work conducted by others using the technology developed in this research are anticipated.

Presentations

1. Royal Institute of Navigation's Animal Navigation Conference 2004, Reading University. Presented TrackTag Technology poster and brief lecture.

2. Royal Institute of Navigation's "New Navigator Seminar 2006", Imperial College, London. Presented lecture on TrackTag focusing on antenna integration issues for animal tracking tags. Resulted in request for publication in "Journal of Navigation".

Prizes

1. 1st Prize in Vodafone Section at the Annual Parliamentary Reception for UK Engineering Research and R&D, 2006.

Chapter 1: Introduction

1.1 Thesis Motivation

The research conducted was in line with the commercial needs of the company. For this reason, the work was a portfolio of short projects linked by theme. The theme linking all of the research together is that of GPS tagging technology. In particular, the research work was focused on the development of NAVSYS' "TrackTag" GPS tracking algorithm technology.

The basic TrackTag concept had been demonstrated through various short field trials. There were, however, major gaps in the design which required research before implementation into the system. These developments form the basis of the PhD material and include;

1. Antenna Integration. The system, as it was before the research started, was a "proof-of-concept" design. A robust tag design for customer use had to be developed. The subsequent research also required the tag hardware to provide signals of a high quality to enable weak signal detection research. It was also regarded essential that the research used "real-world" signals from the outset so that customers could be used to acquire TrackTag data in the field that would be used in the algorithm research. However, there was significant novel design necessary to allow the integration of a small antenna on to the tag. Chapter 3 discusses a new novel antenna test system that was developed specifically to aid the integration and evaluation of miniature antenna technologies into TrackTag.

2. Signal Detection. Detection of the satellite signals is difficult to achieve in harsh environments and/or with limited antenna performance. The signals transmitted by the GPS satellites utilise the benefits of spread spectrum communication in order to allow these signals to be detected on Earth even though, even in good conditions, the signal received is below noise. With such a limited power budget, and using very short samples of recorded data, the issue of accurately de-modulating the signal was not straight forward. The algorithms are then developed further in order to process weak GPS signals.

3. Time Correction. One of the research (and product) objectives is to offer "long

duration" capability. This refers to studies lasting a year or more. Battery and memory capacity issues were addressed by NAVSYS as a separate exercise. The major issue that remained was how to maintain an accurate measure of time as the on-tag clock drifts. This time error directly impacts the accuracy of the position fixes and over a year the error would grow so large that the system would be impractical. Methods for correcting for the time error were therefore examined during this research.

1.2 History of Animal Tracking Telemetry

Animal tracking has been of interest for centuries. Even some animals themselves depend on some form of tracking system in order to hunt their prey. This, of course, includes man and his ancestors who relied on their animal tracking abilities to hunt for food. There are many different senses animals utilize to locate their prey and as they evolved their key senses used to track their food have been optimized. Bats, for example, use acoustic echolocation to track their insect prey [1].

Even now, in the 21st century, the tracking of animals is of great interest although the techniques employed are far more technically advanced. So, what are the motivations behind animal tracking? There are four aspects of wildlife that researchers are most often interested in;

1. Physiology
2. Movements
3. Resource Allocation
4. Demographics

The location of an animal, or group of animals, is fundamental to understanding all four aspects. Whether it is used to explain why the animal's physiology changes in a particular area or interaction between groups of animals, location is always a vital piece of the jigsaw.

Traditionally, the study of wildlife has been, to a large extent, the remit of biological research with pure objective of furthering mankind's knowledge. However, there is growing interest in the commercial sector as society becomes more environmentally aware. A good example of this commercial interest is the study of birds around airports. Even back in 1976 there were concerns over the effects of artificially lighted towers at airports on nocturnal bird migration [2]. Of far more interest to the commercial airlines though, is the behaviour of birds around the runway. This is not due to any concern over the safety of the animals but more down to concerns over possible "bird-strike" causing considerable aircraft downtime and possibly even danger to passengers.

In the Author's opinion, based on feedback from Navsys customers, another recent development provoking the interest of commercial business is the appearance of more and more wind-farms. Of course, a large part of the reason behind building wind-farms is politically driven by environmental concerns. Those same environmental pressures often force the planners to adopt a sympathetic stance with respect to, say, local endangered species of eagle in Scotland. The planners often have to carry out a study into the effects a large wind-farm could have on the local wildlife that frequent the proposed location and may even have to modify their plans to suit the resident wildlife.

Whatever technique is employed for tracking animals there are a few considerations that tend to present technical challenges;

1. Weight/Size
2. Packaging for particular environment
3. Data retrieval
4. Accuracy
5. Duration

These challenges could be tackled individually but the key to animal tracking is achieving an appropriate balance for the application/species in question. For example, a device to be deployed on a pigeon for a week is likely to be far smaller than that deployed on a blue-whale. The device on the whale, however, is likely to have a requirement to last for many months only taking positions every few minutes/hours. So, there is a wide diversity of potential requirements that depend on the individual research being carried out.

The beginnings of what we would consider animal tracking technology were founded in the late 1950's although there have been studies published based on the use of aircraft alone for tracking pigeons [3] by sight in the early part of the same decade. As with many technological advances around that time, animal tracking telemetry was borne out of the space race. A short time after the development of the transistor, radio-telemetry of physiological data taken from pilots during the early days of aerospace research pushed advances in telemetry technology. It wasn't long before telemetry was used to study wildlife, and in particular the heart-rate of chipmunks [4] and both heart-rate and

wing beat from mallards [5]. These represented the first forms of telemetry although they did not have the objective of tracking the creature's position.

This first form of electronic positional tracking was known as "Biotelemetry triangulation" but is usually now referred to as "Radio-Telemetry". It describes a system whereby the animal is tagged with a transmitting device and it is up to trained personnel to work out where the signal is coming from by use of receivers. Some radio-telemetry systems are fairly crude in that they emit an audible tone which changes depending on how strong the signal is. Other, more sophisticated, systems employ some timing information in the signal and can therefore perform a distance calculation as well as direction. The location is therefore determined through estimation of the angle and distance to the transmitting tag. It is, of course, a relative measurement referenced to the receiver(s) position.

The 60's saw some innovations in the use of radio-telemetry technology for tracking animals and its widespread use enabled many advances in animal tracking. Pigeon tracking became a hugely important field of research and 1966 saw the publication of a study based on data collected by use of airplanes and radio-telemetry tagging [6]. That research coupling the new radio-telemetry methods with aircraft highlights a major problem with radio-telemetry that persists today. The range requirements implicit in tracking using a localized communications link make the process of tracking an animal a very difficult and often prohibitively expensive business.

All radio-telemetry systems suffer from the same major problems;

1. Transmitted power level restricts the system range.
2. Requires highly skilled personnel out in the field for entire study.
3. Use of RF spectrum often problematic in terms of licensing.
4. The physical size of the antenna required for optimal RF transmission can be relatively large.
5. Optimum positioning of an antenna on the animal is often not practical.

The first two problems above are well demonstrated while considering some of the lengths researchers had to go to, i.e. the use of airplanes with radio-telemetry. The fact

that an aircraft was required to get close enough to the tag highlights the problem of limited range. The consideration of how much skill must have been required to accurately record the positions relative to the aircraft and then interpret the absolute position of the animal having taken the aircraft position into account. Highly skilled personnel are required to track the tag which makes this approach both very expensive and problematic if the geography/environment is difficult.

So, radio-telemetry therefore exhibits some major weaknesses, mainly the manually intensive (and subsequently slow and expensive) process of acquiring position fixes and also the inherently limited range. On the positive side, it can offer an extremely small tag size and can give a near real-time position estimate of the animal under observation.

Up to the late 70's there was no alternative to radio-telemetry. The radio-telemetry tag devices are relatively simple and were therefore easily reduced in size over the years. This enabled tracking of ever smaller species. Due to the long history radio-telemetry has, as well as the sheer quantity of radio-telemetry tags still sold (100,000 worldwide [7] in 2004), it is a useful yardstick to use when estimating the animal tracking market and how it is split. A snapshot of the market in 2004 is shown in Figure 1 which indicates that there is still a place in the market for VHF radio-telemetry. They are most notable for being the smallest/lightest possible tags due to their simplicity and therefore lend themselves to bird studies.

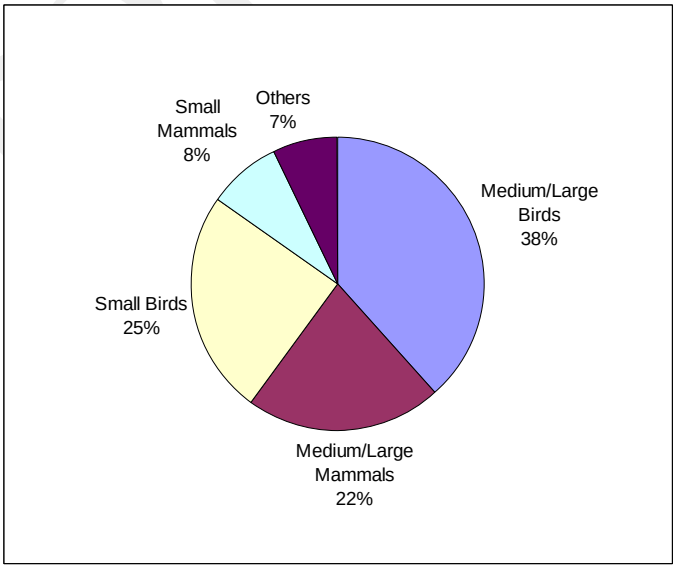


Figure 1. 2004 VHF Radio-Telemetry Market Spilt

From the late 70's there were evolutionary improvements in radio-telemetry which meant they could be made smaller, more efficient and generally more capable, much like electronics in general over the same period. There were also some revolutionary and disruptive technologies on the horizon that would gradually start to compete with radio-telemetry in the animal tracking market. Those new technologies are discussed in section 1.4.

1.3 Animal Tracking Requirements

There are a number of fundamental challenges faced by groups wishing to conduct animal tracking studies. The major requirements placed on the animal tracking tags are now outlined.

1.3.1 Tag Weight

There are two important points to note while considering the impact of tag weight on an animal;

1. A heavy tag can cause the animal discomfort, perhaps even pain. In extreme cases a tag's weight could play a part in the animal's death. The tag weight could tip the balance in nature's battle for survival. It is therefore considered important to have guidelines for weight in order to protect animals in general. These are legally enforced in many countries, including the UK.

2. A heavy tag could change the behaviour of the creature under observation. A classic example would be birds flying around hills instead of over them simply due to the fact that they have been weighed down. Ideally any research conducted on animal behaviour would be able to assume the tag's influence was negligible.

Most, if not all, biologist's research that involves tagging animals now refers to a study conducted in 1971 [8] that offers a guideline for the weight of a tag. It stipulates that the tag should not exceed 4% of the animal's body weight. There is an overwhelming consensus amongst the "bio-tracking" community that these guidelines are suitable and any deviation from them would mean the research would carry a significant risk of being discredited.

Other than the "4% rule", as it's often called, it is a case of what works best for the individual research. Animal attachment methods all vary depending on species, i.e. harnesses are not suitable for animals with a great need for flexibility, like cats for example, but are usually acceptable on birds. Seals often have quite a large tag glued to their forehead [9]! It should be noted that these tags, although large, do meet the "4% rule" and they eventually fall off as the seal's fur moults away naturally.

The influence of tagging is a critical area of bio-tracking research and there have been many studies looking at various tags and on many different species. One example studied the behaviour of Chamois (a relative of goats) having been tagged with a particular radio-collar [10]. The variables under observation included grazing (head down, head up), alert, scratching, licking and how far they were from other Chamois. That particular study demonstrated that there was no observable difference in behaviour when wearing one of the tags in question.

The Chamois study was undertaken to prove the experiment setup (i.e. tag design) was fit for purpose. The study involved 9 tagged and 40 non-tagged animals over a period of more than 2 years. So, there was considerable effort spent, up-front, before the real business of data gathering for a particular biologist's area of interest. This research into how well the tracking devices work on the animal is very common and was even evident during the course of this research described in this thesis.

Weight is therefore extremely important to the design of a tag. It is fairly obvious that reducing the tag weight is a route to creating more demand for the product. Technologies are evolving and enabling smaller and smaller tags to be designed. Radio-telemetry tags lead the way in terms of weight and are still evolving and pushing into smaller animal studies. One recent example is that of the tagging of the Lesser Horseshoe Bat [11], which, due to weighing in at only 4 to 8 grams, was previously unsuitable for any form of radio-tagging. A new radio-telemetry tag was developed weighing only 0.35 grams and a study was then possible.

The power consumption of the tag is the most critical of areas in which the electronics design affects the weight of the tag. The more power required of the tag's battery for each position fix, the more energy capacity the battery is required to offer. The laws of physics dictate that larger capacity batteries mean bigger and heavier batteries. By minimising the tag power usage, the potential tag weight is also effectively minimised. There have been tags developed that integrate solar cells to enable the recharging of a battery. This approach, however, still relies on the tag being very low-power as the amount of energy delivered by the solar cell is proportional to its surface area (and weight).

To cater for as many species as possible, a tag design often incorporates the ability to offer a selection of variants of the same basic design – a tailored approach. As the battery is such a large part of the tag weight, a tailored approach may offer a selection of battery options, one of which may be only just capable of supplying enough energy for the expected trip duration and no more, in order to minimise the weight. The user may even choose to compromise on the number of position fixes expected in order to further reduce the size of battery required.

1.3.2 Tag Packaging

The size and shape of the tag has to be suitable for the species. For example, a long, narrow design may be good for fitting between a bird's wings, but it would be the wrong shape for fitting around a cat's neck. The tag packaging is therefore a critical area of the overall design. While it has to protect the electronics from potentially hazardous environment (for many months in some cases), it also has to balance that against adding too much weight.

Packaging can therefore vary tremendously, depending on the proposed environment for the tag deployment. At one end of the scale there are applications such as tracking small birds where the primary concern is to keep the weight down. The solution may only be "rain-proof" and involve nothing more than a thin layer of plastic wrapping, but it could be the only option assuming it is good enough to protect the tag under any likely conditions.

At the other end of the packaging scale there is the marine environment where packaging often plays a critical role in the success of the tag's overall design. With some marine animals diving to extreme depths the pressure-proofing becoming a major issue. Weight tends to become less of an issue although some species, such as penguins, can push both weight constraints and pressure-proofing requirements.

As tags usually involve some form of electromagnetic transmission they will require an antenna. As the frequency of any transmission determines the optimum physical dimensions, this can often present a requirement for an antenna that is relatively large and awkward to package while meetings size/weight constraints. Packaging materials

can also "de-tune" an antenna as it places a dielectric load in the antenna's near-field. This is a particularly difficult challenge for marine tags where it is necessary to pressure-proof using a relatively thick packaging material that forms a continuous barrier surrounding the entire tag.

1.3.3 Data Retrieval

Whether the tag calculates its position internally, as is done within a conventional GPS tag, or if it is done externally, as with a radio-telemetry-based system, information has to be sent back to the user. The only difference being where the position estimation/calculation is performed. So, there is always the challenge of getting a signal back from the tag.

Transmitter Only Tag

There are many methods that have been employed for getting signals back from the tag. The most basic tag is one that is only a transmitter. This is the case for radio-telemetry systems where the tag would periodically transmit a signal that is used by receivers that estimate the signal strength and direction (or triangulate through use of multiple receivers). It is also the approach taken by the Argos System (discussed later) where the tag signal is received by satellites and thus offers near global coverage.

The tag transmitter solution to data retrieval is certainly an excellent way to get the data back to the user but it has drawbacks;

1. There may often be no receiver within range and power is therefore wasted.
2. The amount of actual data that can be transmitted is very limited.
3. The power required, and therefore weight, scales with range (not using Argos).
4. The number of fixes achievable per day can be very limited (using Argos).

Receiver Only Tag

Many tag designs approach the problem of data retrieval by simply assuming the tag will be collected at the end of its deployment. This is commonly referred to as an "archival tag". It therefore does not actually offer any technical solution to data retrieval, but this is often suitable for certain species. For example, many species have

an element of predictable behaviour, such as nesting, where the biologists can be confident of being able to re-capture the animal to retrieve the tag.

The advantages of this approach are that the tag can store huge quantities of data (not necessarily restricted to positional) as it is not restricted by the bandwidth of any communications channel. It can also be extremely efficient in terms of power as there is no requirement for transmission over distance.

There is one major problem with such an approach – tag retrieval is necessary before any data can be extracted. So, there is a fair amount of risk taken in that the tag could be lost as well as the data it was collecting. No real-time updates are possible either and therefore you could have the case where researchers have tagged animals for a trip duration exceeding a year (or more) and they will have no idea where the animal has gone (and whether it is even still alive) until they successfully re-capture the animal after a very long period of time. This problem is often addressed by adding another simple radio-telemetry-style beacon to enable the researcher to periodically estimate the animal's position. That is only an option, of course, if weight constraints allow for it.

Receiver/Transmitter Tag

There is a subset of tags that aim to provide a compromise between the transmitter and receiver only options. By employing receiver only positional estimation (usually GPS), they transmit the position estimate (or partial calculation) over a communications channel.

There are a few examples of this on the market where a conventional GPS receiver is supplemented with an Argos transmitter. These can be very small although they are still very limited in terms of the number of positions that can be sent back because the Argos bandwidth is very restrictive. The Argos transmitter also requires significant power which places more demand on the battery and could therefore also impact size and weight.

One good example of innovation here was done at SMRU (Sea Mammal Research Unit) at St Andrews University. A conventional GPS unit was integrated with a GSM chipset for use on tags for Seals in the North Sea [12]. When the phone network was available,

the tag would use SMS messaging to send blocks of data back to the team. Although only a small fraction of the deployments successfully sent the data back at the expected times it did show the technology was feasible. There are a couple of major disadvantages of this approach when considering it for wildlife tracking in general;

1. It is a heavy solution. The SMRU tags, for example, weigh around 450 grams.
2. It only works where you have a GSM network.

1.3.4 Accuracy

The accuracy of the animal tracking system employed is an important factor that has to be quantified from the outset of any animal-tracking study. Positional data is used by biologists in order to determine things such as movement and "home range" estimates. These provide information such as how the animals use the environment and migration patterns. "Home range" estimates aim to quantify the area used by the animal and can provide information on the need for certain habitat types.

In all of these estimates, knowledge of the accuracy of the positional data on which they are derived from is crucial. Whether or not the data has sufficient accuracy is ultimately dependent on the requirements of the study in question.

Radio-telemetry triangulation error is a function of the number and accuracy of receiving stations, and an animal's location relative to receiving stations [13]. Radio-telemetry accuracy is, of course, also highly dependent on the user, i.e. how well trained the field personnel are. It is very much dependant on the distance between transmitter and receiver as a major component of error is due to the angular measurement. The absolute error due to the angular error will grow in proportion to range.

Researchers usually, therefore, have to conduct field trials in order to estimate the accuracy of their chosen system in the environment in question (environmental conditions will affect the signals and therefore the system's accuracy). The tests may also be required to find out what range can be expected. The tests should also be done using the same personnel who are expected to be undertaken the data gathering for real as they are part of the system themselves.

One interesting study [14] looked at the accuracy of radio-telemetry under two conditions. The first case being under the usual accuracy test scenario where personnel understand that an accuracy test is in progress. The second accuracy test was performed while the tracking personnel had no knowledge their system was under scrutiny. This "blind study" highlighted a big issue with radio-telemetry in general as the performance of the system offered notably better accuracy when the field personnel knew they were undergoing an accuracy test. The study called for accuracy tests to be done without the tracking personnel's knowledge; however, it does highlight the "human error" factor that is inherent in such a labour-intensive system and one that relies on non-trivial skills.

The accuracy of the receivers themselves, assuming field personnel account for no error in the system, is defined by the accuracy achievable in estimating the transmitter's bearing relative to each receiver. A major problem in estimating the bearing occurs with signal "bounce". This describes the situation where the signals do not follow a Line-Of-Sight ("LOS") propagation path. The signal therefore encounters variables such as topography, vegetation and weather conditions that can all result in the apparent signal direction altering. A study that examined the bias errors in radio-telemetry [15] found that the mean bearing error during their analysis was up to 3° for 68% of the readings, 3-10° for 12% and >10° for the remaining 20%. The resulting positional error is then a function of the system's range and also the geometry of the receivers themselves.

The accuracy of radio-telemetry can be tuned for the application, i.e. the number of receivers could be increased and/or located in strategically important positions in order to offer increased accuracy in certain areas. The expectation seems to be around 100-150m accuracy when using a well setup triangulation system with suitably trained personnel. One example of the assessment of accuracy [16] concluded that their 90% confidence error polygon (the area within which the true location is 90% likely to reside) measured 5 hectares. This figure was determined while using receivers exhibiting relatively small bearing variations and spaced 1.79km apart. A circular error area of equivalent size would result in the positional error estimate of 112m.

1.4 Animal Tracking with Global Navigation Technologies

The history of animal tracking radio-telemetry and the challenges it (and any competing technology) faces have been discussed thus far. However, this research aims to develop technology that will enable animal tracking on a global scale. How the innovations in global navigation technologies have so far impacted the science of animal tracking will now be explored.

The previous sections highlighted the challenges faced in animal tracking applications in general with reference to radio-telemetry systems in particular. This section will now introduce global navigation technologies that could be utilized to track animals. As the overall research goal is to develop technology for global tracking, radio-telemetry is not considered appropriate due to its inherent localised function. It should, however, always be noted that radio-telemetry is likely to be the benchmark against which any new technology will be compared to most of all in the animal tracking market.

1.4.1 OMEGA Background

The first worldwide radio-navigation system was the OMEGA system [17]. OMEGA is related to the Loran (A and C) systems which were developed around the same time using similar technology. However, Loran-A/C was never developed as a global solution. The OMEGA system's initial objective was for near global coverage and accuracy to 1nm (nautical mile) [18]. Its primary application from the outset was for U.S. Navy navigation – in particular submarine navigation.

A result of World War II research, it was based on Very Low Frequency (VLF) signals transmitted from a total of eight stations strategically placed around the world. The VLF signals were broadcast in a frequency band covering 10-14kHz, a range that exhibits significant skin depth into seawater – hence the interest of the system to submarine applications.

The transmitter stations were themselves non-trivial feats of mechanical engineering as they involved large antenna that would either span across valleys or involved the construction of towers that exceeded 1,000 feet in height. As they had to be placed around the planet, the system relied on other countries, such as Liberia, Argentina and Japan, to provide the locations required for global coverage.

The method for determining an OMEGA position relied on phase comparison between signals from at least 3 stations. It was therefore vital that the phase of each station signal was both stable and predictable. The relative phase information of the signals could therefore be used to triangulate a position based on the known locations of the signal sources.

The triangulation process involved finding the hyperbolic lines of position (LOP) created by the stations. The LOP between 2 stations, for example, is made up from the points between the 2 stations that satisfy the phase distances. Unfortunately, there was a complication due to phase ambiguity. As the VLF signals had a signal wavelength of around 16nm, the baseline distance (distance as drawn by a straight line between stations) between LOPs would only be around 8nm as the 2 signal's relative velocity is twice that of a single signal. Bearing in mind that the stations would be placed at distances of many 1,000's of miles apart there were many potential LOPs that satisfied the phase measured. In order to deal with the ambiguity, the receiver would have to be seeded with the correct position accurate to within half of the potential ambiguity, i.e. 4nm.

The system was under trial from 1960 but did not become fully operational until 1971. Before becoming fully operational there was further research conducted by the U.S. military that showed the system to be feasible for aeronautical navigation. It has been described as "what there was before GPS" [19] which highlights how important a role the OMEGA system played in the development of navigation technology. It also played a key role in the following record-breaking milestones in aviation – all occurred in 1987;

1. Richard Branson's first crossing of the Atlantic in a manned hot-air balloon, the *Virgin Atlantic Flyer*.
2. First circumnavigation flight by a light airplane, the *Artic Tern*.
3. First non-stop, non-refuelled flight around the world by *Voyager*.

These breakthroughs would not have been possible without use of OMEGA. OMEGA operated for over twenty-six years and was terminated in 1997.

1.4.2 OMEGA in Animal Tracking

There was some work done that studied the feasibility of using the OMEGA system for this application [20]. The potential for animal tags based on the OMEGA system was considered. However, the basic problem of getting the position data back to the researcher was a significant challenge. There were four methods suggested for getting the data from the tag;

1. Use of a low-data-rate LEO Satellite channel such as SAFIR [21]
2. Use of the data service provided by Argos (much cheaper than positional service)
3. Use of VHF communications channel
4. Archival storage until tag is retrieved

It was recognised that OMEGA could provide a useful method for position determination due to the fact that it could perform reliably in harsh environments and the service was free to use. The accuracy possible, at 1 nautical mile [18], could be considered very poor, but may have suited some studies where long distances were involved. The accuracy could be improved if utilising differential OMEGA for localised studies.

There is no evidence of the OMEGA system having been used for animal studies in the past and with it being decommissioned in 1997 the opportunity has passed. The study conducted did, however, raise some important points about the challenge of extracting the data from an animal while in the field and also the harsh environments in which a tag would have to perform.

1.4.3 TRANSIT

The Transit system [17] was the first satellite-based navigation system. Described as the "Genesis of Satellite Navigation" [22] by two of the researchers involved right for the beginning of TRANSIT development, its concept was formulated in the early 1950's. Ironically, it was the surprise launch of the Russian "Sputnik I" (first artificial Earth satellite) that prompted two relatively junior engineers at the Advanced Physics Laboratory, APL, to study the Doppler effects on the signal emitted from the Russian satellite.

Their work focused on researching whether the apparent Doppler shift could be used to estimate the position of the Satellite. The research progressed well and included analysis of techniques for correcting for ionospheric error using Sputnik II's dual-frequency signals. In fact, the research generated such interest due to the accuracy of its satellite position estimation that in 1958 a new U.S. Navy "Special Project" was to be created which would turn the research on its head – by using known Satellite orbits to estimate the receiver position.

Of course, the system did not yet have any satellites and a completely new system had to be devised which would include the satellites, ground stations and receivers. The primary application for this new technology was the Polaris submarine navigation. The requirements of the system were as follows [23];

1. Coverage: Mean waiting time for satellite to pass overhead of less than 4 hours anywhere on the planet. Longest interval of < 24 hours.
2. Position Accuracy: 0.042 nautical miles.
3. Timing Accuracy: 200μs relative to UTC.
4. Reliability: 97% for each satellite and 98% for complete system (including ground support).

The TRANSIT system became operational in 1964 and consisted of 7 low-altitude (1100km) satellites although the number of satellites increased over the years following. The method for determining position was to wait for a satellite to pass overhead (as they were not always in view and the user could wait up to 100 minutes in practice) and then the receiver would continuously monitor the Doppler frequency of the satellite's signal while also decoding the navigation message provided by that signal. The signals transmitted would contain the satellite orbital data and therefore decoding that, together with estimate of relative position between user and the satellite, provided a position estimate. The 2-dimensional accuracy for a stationary user was 25 metres.

Each satellite pass would last around 10-20 minutes depending on its exact trajectory. The system was initially used by Polaris submarines as well as aircraft carriers in order to update their Inertial Navigation Systems. However, as the system gained acceptance,

other marine applications started to use TRANSIT. For example, the North Sea Oil industry was affected by the use of the TRANSIT system to accurately determine the dividing line between Norwegian and Scottish waters – this was worth millions of dollars per foot in terms of the oil under the sea.

The accuracy improvement over OMEGA meant that inter-continent navigational ties could be established. This resulted, for example, in the realisation that the position of Hawaii on maps of that time was 1km in error!

The TRANSIT system was terminated in 1996 without any significant evidence of its use in the animal-tracking community. Its development was, however, a critical step on the road to the satellite-based technologies discussed from this point.

1.4.4 Argos Background

The Argos satellite system [24] has been operational since 1978 and is operated as a collaborative effort between the National Oceanic and Atmospheric Administration, NOAA, and the Centre National d'Etudes Spatiales, CNES (French Space Agency).

As with the TRANSIT system, the Argos system is based on Doppler Navigation estimates. However, the Argos system works in reverse, i.e. the devices (known as Platform Transmitter Terminals) on the ground transmit up to the satellite.

Only one satellite passing overhead is required to gather the minimum of four signals required by Argos to estimate the PTT position. This is achieved by using signals transmitted at a number of times. The signal generated by the PTT is done using a predefined transmit duty cycle (45 to 200 seconds) and with the average pass duration of an Argos Satellite being 10 minutes, only a handful of signals are received on each pass.

The number of satellites in the Argos constellation is very limited. There were only two in the beginning. This severely restricts the time period in which an Argos satellite is in view of a PTT. The polar orbit that the satellites follow results in the frequency of passes overhead varies between around 7, at the equator, and 28 at the poles [25]. The cumulative visibility is also a very restrictive 80 minutes for the 7 passes at the equator.

Argos offers various classes of accuracy, depending on various parameters including the satellite to PTT geometry as well as the pass duration. The best class of accuracy is specified as a 1-sigma (68%) accurate to within 150m. This would be under ideal conditions so, in fact, the accuracy is often in the order of 1km.

A key advantage Argos offers is the capability for sending data from the device to the user on a global basis. The PTTs transmit 32 bytes of data each time they wake. It is the carrier frequency of the communication that is used by the system to estimate position, the data transmitted can relate to sensor data taken on the PTT device itself. It therefore enables a worldwide communications link (albeit one-way) between the PTT and the user.

1.4.5 Argos in Animal Tracking

The work on OMEGA animal tracking recognised that the Argos system had been the "state-of-the-art" technology in the animal tracking field at the time. However, there were major drawbacks of Argos, mainly its inefficient use of electrical power due to the PTT transmitting without knowledge of whether there was an Argos satellite in view. The Argos system was, and still is, expensive to operate for the user. On top of the purchase of the PTT itself, there are service charges which are typically around $20 per day.

Argos has been used extensively for scientific research involving wildlife tracking. West Indian Manatees (Trichechus Manatus), for example, were tagged using the system in between the years 1986-1996 [26]. 83 of the animals were tracked over 439 deployments off the coast of Florida while 8 Manatees were tracked off the coast of Costa Rica.

The method of attachment to the animal involved a floating tag connected to the Manatee via a flexible tether to a padded belt placed around the base of the tail. The tags used incorporated both Argos and conventional VHF radio-telemetry. While the researchers located the animals around twice a week by means of the radio-telemetry, Argos enabled them to find out were the Manatees had been going while at great

distances from their usual locality. Also, the increased frequency, compared to radio-telemetry points, provided more data regarding the areas that were visited most often.

There are many applications that have utilized the Argos system, especially in the marine environment. Indeed, more than half of the Argos PTT units sold are being used to track marine buoys [24]. Other marine wildlife that have been studied by use of Argos include Loggerhead Turtle [27], Pacific Walruses [28] and Greenland Sharks [29].

While many studies that have used Argos are considered successful, they often also serve to highlight the enhanced performance that GPS would offer in terms of the data's spatial and temporal resolution.

The study on Pacific Walruses [28] attempted to integrate conventional GPS into the tag alongside the Argos PTT. This was not very successful though as only around 5% of GPS position attempts actually calculated its position and far fewer than 5% yield was achieved at sea. This highlights the difficulties presented to GPS in the marine environment. It was suggested that the lack of successful GPS fixes was due to brief and infrequent surfacing of the Walruses as well as the proximity to cliffs and other animals. Although the Argos element worked as planned during the research, despite sharing the same environment, GPS technology was seen to be far less robust in that marine animal tracking application.

Argos has not only been used in marine applications as over 10% of PTTs have been used in tracking either terrestrial animals or birds [30]. Even though it is far less capable than GPS in terms of accuracy and availability, it does offer one major advantage. That is the ability for the tag to transmit its position back to the user. This could be either the relatively poor quality Argos estimate or the payload data that is sent back could contain GPS position data. This can often provide a solution to data retrieval from a tag that could travel many 1000's of miles during a deployment.

1.4.6 GPS Background

In 1995 the GPS system was declared fully operational [17]. Like most major technological advancements in navigation, it was the U.S. military that funded the

development and were the primary users. The GPS system was designed with the aim of providing instant location fixes anywhere and at any time. This relied on a number of new technologies, namely;

- Stable space platforms in predictable orbits
- Ultra-stable clocks
- Spread-spectrum signalling
- Integrated Circuits

There are over 24 satellites in orbit in the GPS system; however, there is also the Russian equivalent, GLONASS, and the soon to be launched, Galileo system from Europe. GPS technology is now ubiquitous and is used in many and varied applications – from military aircraft navigation to civil engineering surveys.

The use of GPS in so many applications has brought the cost of the receivers down to consumer levels. Indeed, there have been small, hand-held, inexpensive GPS units available to hill-climbers for years now. The accuracy of GPS receivers varies tremendously, for example a cheap hand-held unit aimed at the hill-walkers market may have a <100m accuracy specification whereas, GPS surveying equipment is often down at the 1cm level.

The system is based on the principle of "time-difference-of-arrival", TDOA, between signals received by the device from a number of GPS satellites. The relative time measured between signals is used to calculate the position. This relies on accurate estimation of satellite orbits and also accurate estimation of time.

The information required to calculate satellite orbits is transmitted by every satellite along with, time information and some system correction factors. This data is transmitted using a spread-spectrum modulation scheme which also means that every satellite can transmit at the same frequency and the receiver can receive multiple satellite signals simultaneously.

1.4.7 GPS in Animal Tracking

GPS has been used in many animal tracking studies and is becoming more and more prevalent in the bio-tracking research community. For example, there has been great interest in tracking pigeons for many years and there are now more and more studies being conducted that are based upon the use of GPS tags on pigeons [31-35]. It has become the de-facto standard method for determining position in so many applications due to its accuracy and near constant availability [36].

Home-range size estimation of species is often studied in wildlife biology to assess habitat relationships. Use of GPS to accomplish this has enabled this to be done more easily and more accurately as it does not depend on skilled researchers in the field keeping constant track via radio-telemetry systems, often in very difficult and remote terrain, for example while studying Moose at high latitudes of the Canadian tundra [37].

A study of the territorial behaviour of Lesser Spotted Eagles [38] also highlights the benefits that GPS has brought to the research of home-range information. Seven eagles were tagged for a single season and almost 3,000 fixes helped determine their home-range which spanned up to 172km^2. Carrying out a similar exercise using radio-telemetry would be prohibitively expensive if even possible.

There is also more interest in interfacing physiological data to the tag in order to study how animal physiology changes in different circumstances. Commonly referred to now as "Biotelemetry" [39], it is the remote detection and measurement of an animal function, activity or condition. Having constant availability of the position data is essential in order to study biotelemetry as a correlation between physiology and position (and perhaps velocity) is a time-critical task. There was some research done that involved integrating EEG (electroencephalography) with GPS on pigeons [40]. There are other applications that have involved the integration of other data into GPS positional data. Sea depth, for example, while studying marine animals, can offer more data in terms of where the creature goes while it is underwater.

These additional datasets can often add to the problem that there has to be a method for extracting the data from the tag. This is sometimes addressed by use of Argos. However, that has a very limited bandwidth. Another method for transmitting the tag

data has been by use of the mobile phone network. This has been successfully demonstrated while tagging gray seals (*Halichoerus grypus*) in the North Sea [12] as previously discussed. The GSM enabled tags automatically sent SMS text messages once every two days. Haul-out data and coastal location data were incorporated into the messages. Data was presented for a three-month period. The obvious problem with this technique is that it will only work within the bounds of a GSM network – hardly global! Also, the power requirements were such that the tag had to incorporate two D-cell batteries and was therefore large.

There are some major challenges faced when designing a GPS-based tag for animal tracking. By its nature, GPS technology does not perform well in harsh environments. The signals it must receive are hidden below noise and conventional GPS processing often relies on the reception of uninterrupted signal for many seconds before a position fix can be determined. This presents major problems in areas such a dense forestation or the marine environment where "wave-wash" poses a significant hurdle to continuity of received signal.

The performance of GPS under forest canopy has been studied while using GPS collars on Elk in South Dakota [41]. The GPS performance was evaluated over various categories of vegetation and topography. The probability of a failure to calculate the collar position was found to be near zero while under forest canopy of less than 40% cover. The probably of failure then rose to around 15% while the collar was under a canopy of between 41-70%. With forest canopy cover greater than 70% the chance of failing to calculate a position rose to almost 50%! In practice these figures are probably optimistic as the receiver was given 90 seconds to acquire every time. Long-duration tags are unlikely to have the luxury of such a long ON time to gather the required signals and so the effects of forest canopy on GPS are very significant.

The problems encountered while using GPS at sea have been also been a challenge for research groups working on autonomous submarine design. One such group at the Dunstaffnage Marine Laboratory, Scotland, looked at the problems posed by the use of GPS in the marine environment [42]. It was highlighted that the usual cause of failure is due to wave wash-over of the antenna. As GPS processing requires around 45 seconds of uninterrupted signal reception, occasional breaks in the signal makes the

decoding of signals very difficult. The team at Dunstaffnage devised an algorithm for piecing together the fragments of the signal, as they were acquired, in order to obtain the signal length required for a position fix. Although this proved to be successful through some simulation tests, it does involve some intensive processing and relies on relatively long signal acquisition times due to the receiver having to wait to receive sufficient fragments.

1.5 Summary of Current Animal-Tracking Technologies

Conventional radio-telemetry, Argos and GPS stand out as the technologies that have featured most heavily in tracking animals. Each of these has particular strengths and weaknesses that are summarized in Table 1. The main points to note are that;

1. Satellite-based systems suffer badly when used in harsh environments. These could be marine (i.e. submersion and/or wave-wash issues), areas of poor visibility due to tree-canopy coverage or perhaps even the orientation of the tag changing due to the animal's movement.
2. Only Satellite systems can offer very wide (or global) coverage in practice.
3. Radio-Telemetry and GPS offer good accuracy and position fix availability.
4. Retrieval of positional data from GPS tags is a major challenge.

The conclusion from this chapter is that (VHF) radio-telemetry technology is what the scientific community has grown used to over the decades. The technology is still constantly evolving although it will always suffer from being an inherently localised tracking system. As the purpose of this research is to develop global tracking technology, radio-telemetry is not a suitable candidate for study.

This research now concentrates on GPS technologies as they have the potential to offer accuracy surpassing what the "Bio-tracking community" are used to with radio-telemetry. It is worth noting, however, that any GPS-based tagging system will usually be compared, by the bio-tracking community, against the corresponding performance offered by radio-telemetry. The major benefit GPS offers is that it offers truly global coverage. There are, however, major challenges to getting GPS tagging technology to operate in many animal tracking environments. These challenges form the basis for the research contained in this thesis.

System	Strengths	Weaknesses
VHF Radio-Telemetry	1. Simplest design = potentially smallest possible tag. 2. Tag retrieval not necessary. 3. Proven technology in animal tracking market.	1. Skilled field personnel required = expensive 2. Accuracy good but also depends on "human factor". 3. Only localised coverage.
Argos	1. Easy to use. 2. Position fixes are sent to user in real-time. 3. Tag retrieval not necessary. 4. Global coverage.	1. Accuracy is poor. 2. Position assumed at sea-level. 3. No ability to estimate velocity. 4. Very infrequent position fixes. 5. Transmission to satellite can be compromised by environment. 6. Service charges makes it costly for long-duration studies
Conventional GPS	1. Easy to use. 2. Fine temporal resolution - constant availability. 3. Fine spatial resolution/accuracy (<100m). 4. Global coverage. 5. Cheap to operate.	1. Requires relatively long and uninterrupted exposure to satellite signals (usually 15 to 120 seconds). 2. Short battery life due to time and processing power required to compute each position fix. 3. Complex processing on-board makes unit relatively large. 4. Satellite signal can be greatly compromised by environment.

Table 1: Animal-Tracking Technology Comparison

Chapter 2: GPS in Animal Tracking

2.1 Key Factors in GPS Design

2.1.1 Antenna design

Patch

There are several types of antenna used in GPS systems today. By far the most commonly used type is the "Patch" antenna. This is the case across both the main consumer markets for GPS as well as the specific area of interest – animal tags.

A "Patch" antenna comes in various sizes and is formed by placing a rectangular plate a certain distance from a ground plane. The exact diameters of the patch edges and plate to ground separation tune the antenna to the frequency of interest, the plate dimensions being approximately half of the wavelength required. Ceramic dielectric is often also used in between the plate and plane in order to miniaturise the physical dimension required for the given frequency by reduced the effective dielectric constant [43].

Patch antennas offer excellent gain where the orientation of the antenna can be assumed to be pointing skyward. This is due to the antenna gain pattern being highly directional and the design of the antenna matches the polarisation of the GPS signal, i.e. RHCP (Right-Hand-Circularly-Polarised). However, the tuning can be greatly affected by size and, to a lesser extent, shape of the ground plane. Also, detuning can occur when dielectric material is placed in the antenna's "near-field".

Quadrifilar Helix

An alternative to the patch is offered by the "quadrifilar helix". The helix is constructed from two spiralling loops around a dielectric core. This forms a spinning dipole across two orthogonally phased loops, creating the desired RHCP polarization with a broad beam width far field pattern and a null pointing toward the bottom of the antenna.

There a two major advantages of the helix design over that of the patch; robustness against near-field de-tuning is a particular advantage of the helix design [44, 45]. It was designed with mobile phones in mind where the placement of both the user's head and hand in close proximity to the antenna have often proved troublesome in designing GPS functionality into the mobile phone.

Another advantage of the helix design is the broad beam-width it provides. The gain at low elevations (assuming the helix is mounted vertically giving the optimum gain directly overhead) is significantly better than that provided by a patch antenna. However, the optimum gain is not as good as the patch. The helix antenna is therefore less directional and in some applications, where perhaps the orientation is not very predictable, this could be preferable.

Linear Antennas

The simplest antennas can be constructed by use of a wire of certain length (λ/4). Known as monopoles, they do not have the desired RHCP property and therefore immediately lose half the power. They are, however, extremely efficient in terms of gain versus size/weight. The gain pattern has in theory a very sharp null in the direction of the antenna axis.

"Chip Antennas" have been developed which are similar to monopoles only they employ some dielectric material in the near-field in order to reduce the effective wavelength and hence to dimensions required.

Summary

It is envisaged that antenna selection will be critical to the success of the GPS tag. As the first component in the signal processing chain, any unnecessary losses at the antenna will feed through the entire system. It is therefore essential that the performance of the antenna is well understood and the antenna selection is appropriate for the particular animal study in question. For this reason, the tag hardware should be designed to be adaptable and allow for the integration with a range of antennas of differing fundamental types.

2.1.2 Signal Detection

The process of signal detection provides the range measurements required to triangulate a position fix. Demodulation of the BPSK signals provides a measure of the code phase of that signal and therefore the pseudo-range to the corresponding satellite. The pseudo-range is therefore essentially a by-product of detecting the signal. As the code repeats

every millisecond, a complete code cycle covers approximately 300km. So, if the code-phase is known, the range is known, modulo 300km.

The GPS signals, transmitted from each satellite, consist of two radio frequencies, L1 and L2 [46]. [In fact, there are a further two frequencies (L3 and L4) associated with the US military and are classified]. This thesis is restricted to the civilian signals, which are transmitted on L1, and their characteristics are as follows;

1. Carrier Frequency of 1575.42MHz
2. BPSK (Binary Phase Shift Keying) spreading code ("gold-code"), unique to every satellite, with "chipping" rate of 1.023MHz. The codes are 1023 bits long and therefore repeat every millisecond.
3. 2.046MHz bandwidth.
4. Navigation data. This is the data that is required to compute the satellite positions. It is transmitted at 50Hz.

The BPSK spreading code is a binary sequence which, when used to modulate the phase of the L1 signal, introduces a 180° shift in phase whenever a bit switch occurs in the spreading pattern (gold-code).

If we consider the L1 carrier is multiplied by the spreading pattern, c(t), as shown in Figure 2, the resultant waveform is the same continuous sine wave but with 180° shifts any time the spreading pattern changes sign.

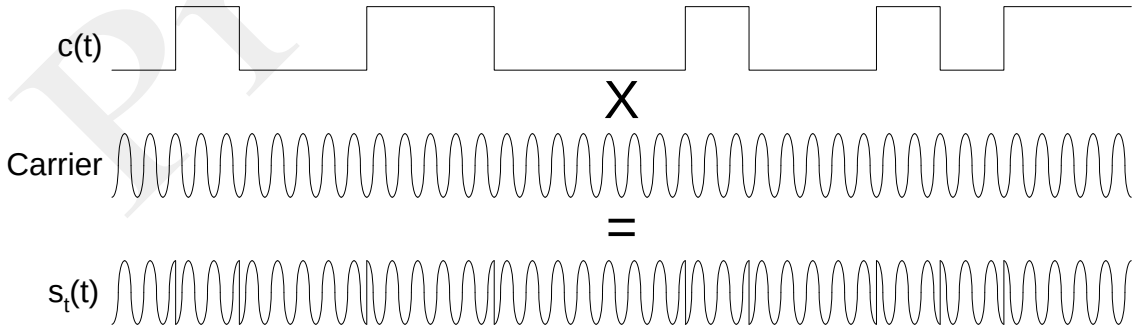


Figure 2. BPSK Spreading

The general BPSK formula (1) describes the spreading mathematically [47]. The signal power is given as P. The spreading code, c(t), is multiplied by a cosine wave with

frequency ω_0 and phase θ_d. The phase term is actually the data and is not considered at this point.

$$s_t(t) = \sqrt{2P} \times c(t) \times \cos[\omega_0 t + \theta_d(t)] \quad \textbf{(1)}$$

The frequency, ω_0, is the carrier frequency, "L1", and is nominally 1575.42MHz. However, the signal is rarely detected at that precise frequency due to Doppler effects caused by the satellite's motion as well as the receiver's reference oscillator offset error.

Gold-Code Generation

In order to detect the signal, the unique pseudo-random bit sequence for the given satellite signal must be used and it must also by delayed such that it is in phase. The CA codes for each satellite are unique 1023 bit long binary sequences [46]. They are generated using a combination of two linear feedback registers (Figure 3) which follow the polynomial equation given as equation (2). The two shift registers are pre-loaded with values according to which satellite PRN code is required.

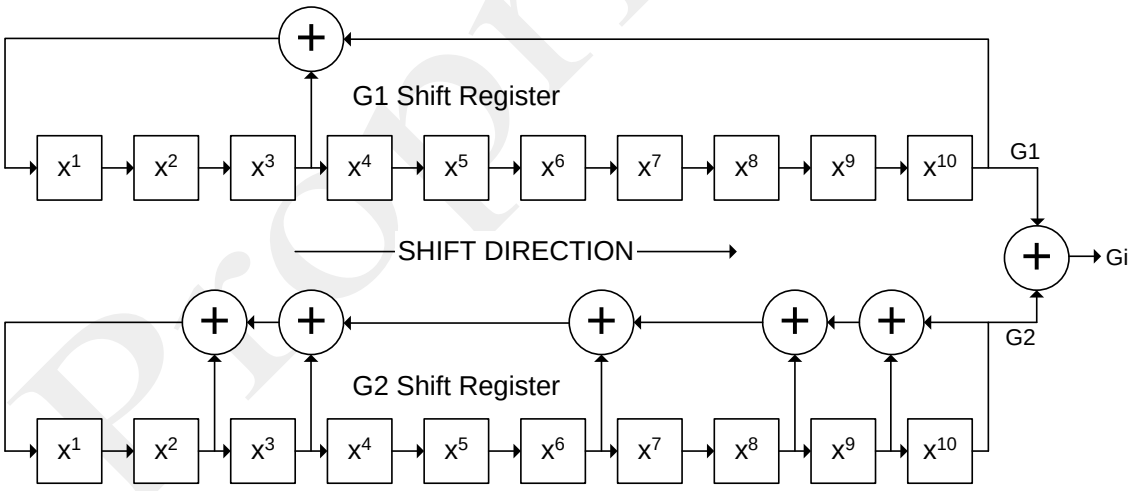


Figure 3. Gold Code Generator

$$G1 = x^{10} + x^3 + 1$$
$$G2 = x^{10} + x^9 + x^8 + x^6 + x^3 + x^2 + 1$$
$$Gi = G1 + G2 \quad \textbf{(2)}$$

Code Stretching – Doppler Search

The objective for the gold-code generator is to create a copy of the signal that is expected from the satellite of interest. This is done by creating the signal as it should look over a time period equivalent to one BPSK period, i.e. 1ms. For example, assuming the signal is sampled at 5MHz, 5000 samples are expected in each millisecond. However, if there is a positive Doppler shift, the resultant frequency will be higher and more than one cycle of gold-code will be received within 1ms. Conversely, if there is a negative Doppler shift, less than one complete gold-code cycle will be expected in 1ms.

The binary sequence expected will therefore be stretched, or compressed, depending on the Doppler frequency experienced. In addition, the imperfections of the receiver local oscillator mean that there will be a frequency offset due to the oscillator error. This simply looks like a common Doppler offset on all signals.

The frequency of the signal as it reaches Earth can vary by around +/-6kHz depending on the velocity of the Satellite [17]. For this reason, "Doppler Bins" are used in order to find the signal at the correct frequency. This often involves many different frequencies being searched before the signal is found. The number of Doppler bins required will depend on the possible oscillator drift as well as the bandwidth of the tracking loops.

Code Search

Within each "Doppler Bin" a search through all possible code phases is required. The "code search" attempts to detect the signal by matching the stretched code with the incoming data stream.

Only when the correct code at the correct phase is used will the signal be detected. The diagram in Figure 4, as well as equation (3), illustrates part of the process involved. The signal coming from the satellite has to match the signal internally generated by the receiver. Correlation of the two signals involves multiplying the corresponding bits and summing the result.

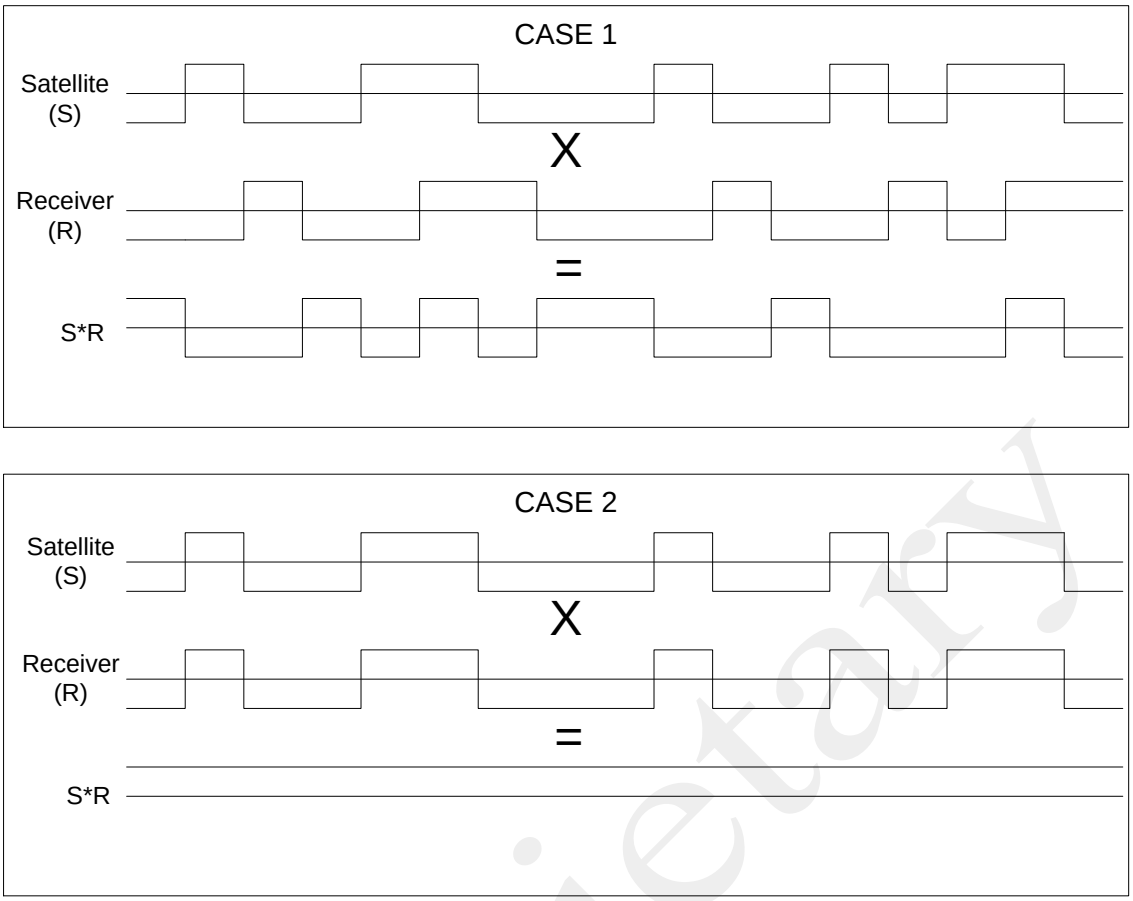


Figure 4. Basic Bit Correlation

$$C_n = \sum_{k=0}^{k=N-1} s_{n+k} \times r_{n+k}$$

(3)

In both cases the signals used are the same but rotated by one bit. By observation, it can be seen that the second set of signals will give a higher correlation value due to all of the bits matching. So, the code-phase required for a match is correct in the second case.

This simple example only uses a 16 bit sequence and therefore has 16 possible bit phases to attempt to find a match. Although the GPS signals have 1023 bits and are repeated every millisecond, the TrackTag front-end, for example, samples this data at 5MHz and there are therefore 5000 samples and 5000 possible code phases to try before finding the correlation peak. It is convolution of the signal against the expected "gold-code" that is required to ensure the signal is acquired no matter what its relative phase is with the receiver code generator.

So, at its simplest, GPS signal detection requires many correlations of signals over all possible code phases and many possible Doppler frequency bins. The output of interest is the code-phase at which the signal's peak occurs.

2.1.3 Receiver Sensitivity

The conventional approach to GPS receiver design involves use of correlators. Correlation is required to "de-spread" the spread-spectrum signal modulation used in GPS. Figure 5 shows the most common top-level hardware design [17].

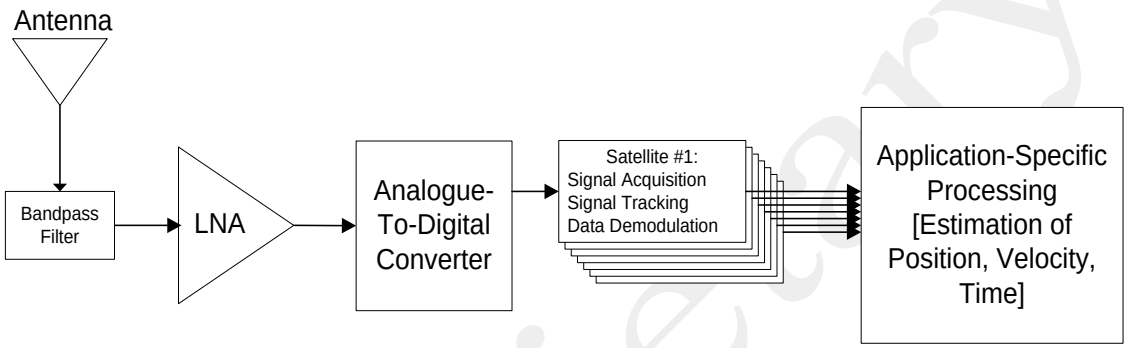


Figure 5. Conventional GPS Receiver Block Diagram

Although the design has one front-end analogue section, a number of DSP sections are required in order to generate enough GPS channels to enable of navigation fix. Each channel has to be dedicated to acquiring a unique GPS satellite's signal. In practice this means one of a receiver's specifications is the channel count – the higher the better.

On the analogue side, all that is done is the received signal (at this point still buried under noise) is amplified and band-limited. It is also mixed down either to an IF (Intermediate-Frequency) or to baseband and then sampled. The down-converted signal is usually filtered again to reduce the bandwidth further with the aim of reducing noise in the system. Although the GPS signal is always below noise-level throughout the analogue section, the frequency spectrum within which the signal will be hidden is amplified and any noise out with that frequency range of interest is greatly reduced.

A major contributor to the actual frequency encountered is the receiver's reference oscillator drift. As the received signal is down-converted using PLLs (Phase-Locked-Loop) there is an offset due to any deviation from nominal on the reference frequency.

This has the effect of shifting the perceived Doppler on all satellites by the same amount.

Signal Power

The receiver sensitivity performance is defined as the ratio of signal-to-noise at which a threshold can be set to determine whether a peak is a signal peak or just noise. Signal-to-noise can be expressed either as a normal SNR ratio (measured in decibels or 'dB') or as Carrier-to-Noise Density Ratio, C/N_0. The C/N_0 term is most commonly used in the GPS industry and is a measure of the ratio between received signal power and received noise density and has the units 'dB-Hz'.

The carrier received from the satellite is expected to have a power of at least -160dBW [48]. This is assuming a direct line-of-sight with no attenuation. That is a level 160 dB below 1 Watt and is more commonly expressed as -130dBm which is simply the same power level but referenced to 1mW rather than 1W. The signal level in the receiver is highly dependant on the antenna's gain, directivity and RF front-end noise characteristics.

The noise density, N_0, expected is given by equation (4);

$$N_0 = kT_{eq}$$
$$T_{eq} = \frac{T_A}{L} + T_0 \cdot \left[\frac{L-1}{L} + (F-1) \right] ^\circ K \quad \textbf{(4)}$$

Where;

k: Boltzmann's Constant (1.38×10^{-23} W/°K-Hz)

T_A: Antenna Noise Temperature.

L: Transmission Loss between antenna and the 1st LNA stage.

T_0: Ambient temperature of RF electronics.

F: LNA Noise Figure

Each GPS receiver will have different values for these parameters but if we take some nominal values of T_A =130°K, L = 1.1, T_0= 290°K and F = 1.259 (1dB), the resulting noise density is -205dBW-Hz. This example would therefore have a C/N_0 of 45dB-Hz when the minimum expected signal power of -160dBW (or -130dBm) was present.

The sensitivity threshold will have to be at a level below the minimum expected signal power for the receiver to be able to cope with any signal attenuation encounter in the real world environment.

Tracking Loops

As the received 'signal' is passed into the DSP side, the 1st thing that is done is the removal of Doppler. This Doppler is a result of the Satellite's motion and can be up to 6kHz either side of nominal [17]. The correlator requires that this Doppler offset is removed and any signal is presented close to the expected nominal frequency.

There is therefore a carrier tracking loop which is required to lock on to the satellite signal's carrier frequency. The satellite carrier frequency will include its own specific Doppler offset (at that moment and at that receiver position) and will therefore be different from satellite to satellite.

The code tracking loop adjusts the code phase to track that of the incoming data stream. Assuming the signal is present, a "correlation peak" will occur at the point where the gold-code and the signal are aligned. This also tells you the code-phase of the incoming signal. It is the code-phase that is of most interest as it provides a measure of relative timing and therefore distance. The correlator output peak falls rapidly – from maximum correlation, down to zero at ½ a chip away (i.e. ½ × 1/chipping-rate).

Finding the correct starting point for code tracking is a non-trivial task and receivers can handle that in many different ways. The most basic would do a purely linear search on each and every. More advanced algorithms would try the satellite expected to be at the highest elevation first and then the others would follow in descending order of elevation under the assumption that there is more chance in detecting the highest satellites. Usually once one signal is acquired, an estimate of the other signal phases is calculated based on the current/last position estimate. An expanding search from that estimate will most often reduce the computational time required for the code-phase searches.

The sensitivity of the receiver depends on all of the components in the front-end RF chain as well as the processing employed. If we assume that everything is done to ensure the antenna integration and front-end matching is optimal, the signal detection routines will define how sensitive the receiver is.

SiRF manufacture the "SiRFstarIII GSC3e/LP" and it is assumed in this thesis to be the benchmark "state-of-the-art" in conventional GPS [49]. It is one of the latest in SiRF's GPS products and is geared towards offering high performance and low power operation. Sensitivity of the SiRF chipset is specified as being -142dBm (around 33dB-Hz) for acquisition. Higher sensitivity is achievable once the tracking loops pull-in and the receiver is no longer "acquiring" but is "tracking" the signals. In terms of the animal-tracking application it is assumed that it is only the acquisition specification that is of interest as once the signals are acquired a position will be logged and the tag will power-down.

2.1.4 Almanac and Ephemeris

The GPS Navigation Message transmitted by the satellites largely consists of the Almanac and Ephemeris data along with various correction factors such as clock bias and satellite health flags.

Both Almanac and Ephemeris data are used to estimate the position of the satellites but they have differing objectives. The Almanac data is a subset of the Ephemeris data and has far less detail. However, it is the same Almanac data that is transmitted on all satellite signals and it contains coarse orbital information for all GPS satellites. This means that the reception of just one signal can provide enough information to the receiver to allow it to predict which of the other satellites are likely to be visible [17].

The Almanac data is used, in a conventional GPS system, to predict which signals are likely to be present and therefore can reduce the time spent acquiring the necessary signals. As the satellite orbits are approximated using Almanac data, their signal Dopplers can also be estimated in order to further reduce the time required to acquire subsequent signals. The orbital positions are not accurate enough for use in the GPS navigation solution.

Ephemeris data for each satellite, in contrast, is broadcast only by the satellite in question. The Ephemeris data offers sufficient accuracy for the navigation solution. The Ephemeris is repeated every 30 seconds and is constantly being updated by the GPS ground segment in order to maintain accuracy.

2.1.5 Navigation Solution

The navigation solution is central to the whole GPS processing system. In a conventional GPS receiver, the navigation algorithm's inputs are;

1. Range information in the form of "pseudo-ranges" to each satellite that has been detected.
2. Current position estimate.
3. Current time estimate.

By using these 3 sets of data a position can be calculated. Explanation of how this is done can be found in Appendix A.

Pseudo-ranges are calculated using the code-phase measured through the process of signal detection (and actually also require the current position estimate). The position estimate is usually the last known position and the current time estimate is the time held on the receiver's RTC (Real-Time-Clock) which can usually be corrected whenever certain parts of the GPS signal structure are de-coded.

The code-phase outputs from the signal detection process are related to the transmission time of the signal from Satellite to receiver. This is the critical relationship that all GPS calculations are centred around. The "pseudorange" to a satellite is this measured transmission time scaled by the speed of light in a vacuum, c, as in equation (5).

$$\rho = c(t_R - t_T) \tag{5}$$

Pseudorange, ρ, is therefore related to the time of signal reception at the receiver, t_R, and the time of transmission from the satellite, t_T.

A more complete equation includes error terms and relates the receiver and satellite positions rather than times. Equation (6) shows how the receiver position, r_u, is related to various parameters;

$$\rho_i = |r_i - r_u| + c \cdot b_u + \varepsilon_i \tag{6}$$

ρ_i: Pseudorange to satellite 'i'.

r_i: Satellite's position.

b_u: Receiver clock error.

ε_i: Composite of all other errors such as atmospheric delays, receiver noise etc.

The standard navigation solution therefore requires at least 4 simultaneous equations to be solved for each of the 3 dimensions of r_u plus the unknown time offset due to receiver clock bias [we can ignore the ε_i term for now as it can only be addressed by

implementing advanced GPS augmentation such as WAAS or DGPS]. This implies that at least 4 satellite signals are required to generate a position fix.

As the pseudoranges are measured by demodulation of the relevant satellite signal, as it is related to the code-phase of the signal, the only other unknown is the satellite position. It is the ephemeris data that is used to estimate the satellite position and this relies on an accurate estimation of the time of transmission. Any error in the time assumed in the equations will result in an error in the calculated satellite position at the time of transmission and therefore a subsequent error in estimated receiver position.

2.1.6 Time-To-First-Fix

"Time-To-First-Fix" (TTFF) is a term used throughout the GPS industry to define the length of time it takes the receiver to compute a position. The TTFF is usually specified in terms of three different scenarios;

1. "Cold Start". This describes the situation where the receiver has no a-priori knowledge of the signal, time or satellite almanac/ephemeris. This would usually be the case if the receiver has not tracked signals for over 8-12 hours.
2. "Warm Start". This scenario describes when the receiver has complete almanac but its ephemeris requires updating. This can be the case if signals have not been tracked for several minutes – depending on whether the satellite orbits have changed during that period.
3. "Hot Start". This occurs when the receiver has relevant ephemeris data and accurate time. Usually the case when the receiver's last position estimate was calculated within the last few minutes.

The latest SiRF chipset leads the market in terms of its TTFF performance offering a Hot Start of <1 second and Cold Starts of under 35 seconds. As mentioned previously, in the animal-tracking application the receiver is most likely to be used to start up "cold" and power down immediately following a position calculation. It would be a severe restriction if the tag depended on a guaranteed good signal tracking every 15 minutes. The specification of interest should therefore be that of the "Cold Start" especially given that an animal could often be under cover for long periods. 35 seconds is the TTFF figure of interest.

2.1.7 Power

The SiRF chipset consumes a mere 62mW which is far lower than most conventional GPS – other SiRF receivers from the same family are around 4 times as power hungry! Of course, this figure covers only the GPS engine and does not include any power required for other supporting electronics.

It is the energy requirement for each position fix that is of concern as that is what can be related to the battery capacity. We will take the SiRF chipset's market-leading specifications in terms of both power consumption and TTFF, 62mW and 35 seconds respectively to give an energy requirement of 9.65 Joules per Fix (running from a 3.3V supply).

2.2 GPS Augmentations

2.2.1 Differential GPS

In order to achieve greater accuracy Differential GPS (DGPS) can be employed. This augmented version of GPS uses a nearby reference receiver at a fixed and accurate position. The reference station can calculate correction factors accurately as it can assume its position to be constant. These correction factors can include satellite clock bias error and atmospheric errors due to the Ionospheric and Tropospheric propagation delays. It can therefore greatly reduce the errors grouped together as ε_i in the pseudorange equation (6).

DGPS, however, requires that a reference receiver is positioned locally and therefore does not offer a truly global solution.

2.2.2 Assisted GPS

The term "Assisted GPS" describes a system that relies on the presence of a cellular phone network that is set up to provide GPS data over the phone network. It was introduced in order to allow the emergency services in the U.S. to pinpoint a "911" caller using their mobile phone's position relayed across the network.

A major problem encountered in attempting this is the so called "urban canyon" challenge. As people will often try to use the service in towns or cities, they will often be positioned next to buildings therefore reducing the probability of having a GPS satellite signal in direct view. In large cities, where the user could be between very tall buildings, the effect of the receiver being in close proximity to buildings is a significant attenuation of the GPS signals – in a similar way forest canopy often attenuates GPS in animal-tracking applications.

The assistance data provided by the network includes both precise time and orbital data. With the receiver having the time and ephemeris information, there is a far higher certainty of where to look for the GPS signals. The normal signal detection threshold of around -130dBm can by improved by 20-30dB [50]. This is achieved as the signal tracking loops are able to search for the signals with tighter constraints and therefore have a higher probability of detecting weak signals. The AGPS capable receiver also benefits in terms of Time-To-First-Fix due to the prior knowledge of key parameters

necessary to acquire the signals. The SiRF chip previously discussed is capable of being assisted via GSM and offers an acquisition threshold of -155dBm (-142dBm unassisted) and TTFF of 6.5 seconds (35 seconds unassisted) while using the phone network data [49].

The Assisted GPS technology would be a very useful aid to an animal-tracking tag as it would offer a method for detecting weak signals. The major deficiency, of course, is the dependency on a cellular network over the full geographical area of interest.

2.2.3 WAAS

Known as WAAS (Wide Area Augmentation System) in the U.S. or EGNOS in Europe, this system is satellite based and provides information on the integrity of the GPS constellation as well as atmospheric correction factors. It has been introduced in an effort to overcome the deficiencies in today's air traffic infrastructure and aims to improve the "integrity" of the GPS system. The aviation community has very strict requirements in terms of the availability, accuracy and integrity of the navigation systems used by civilian aircraft. The "integrity" is a measure of how much the measurements can be trusted, i.e. if a satellite signal is incorrect it must be flagged as such and within a certain timeframe. WAAS is intended to be a worldwide system but is not currently fully deployed.

The WAAS signals are broadcast at the GPS L1 frequency and have similar RF modulation although the data encoding is different. The front-end hardware design for GPS signal detection reception can also detect WAAS as the signals differ only at the digital level. The signals are also freely available and will cover most of the globe. This makes the use of WAAS in animal tags a very attractive option.

Animal tags could benefit from WAAS processing in the following possible ways;

1. Integrity checking on GPS signals.
2. An extra observable made available to navigation solution.
3. Atmospheric corrections to improve accuracy.

2.2.4 Snapshot Processing

Snapshot Processing describes a method by which the signal is detected by taking short chunks of data, "snapshots", and processing each in turn. This contrasts with the conventional approach where a continuously tracking correlator-based is subject to fluctuating signal levels and changing nature of the signal environment.

Snapshot processing allows the use of FFT-based signal detection which performs better with weak signals. The main problem encountered while using a conventional correlator approach to signal detection is that of "False Lock". This term describes the case where a correlator locks on to a noise peak and usually occurs when the thresholds are set relatively low (close to noise) in order to acquire weak signals. The FFT-based approach, on the other hand, does not suffer False-Lock to the same extent as the technique always provides all possible correlation peaks in the signal and therefore a more informed threshold decision can be made. Signal detection is discussed in detail in chapters 4 and 5.

2.2.5 Fastloc™

There is an interesting product that has been used in marine wildlife studies called "Fastloc™" which is designed and manufactured by Wildtrack Telemetry Systems [51]. The Fastloc device is targeted at marine applications and it aims to provide a GPS tag that can handle the effects of discontinuous signal reception primarily due to wave-wash.

Fastloc is a commercial product and it is therefore difficult to get detailed information about how it is designed. However, through discussions with the Fastloc creator, the author knows that the Fastloc receiver doesn't use a standard GPS chipset. It utilizes a variety of available RF front end chips, but the baseband section is a proprietary DSP-based design.

The on-board processing only has to de-modulate the satellite signals to provide the received satellite's code-phase. This snapshot-processing is stored, along with a "rough timestamp", and then post-processed later on another system. The system can be integrated along with ARGOS, GSM or VHF/UHF in order to provide a communications link back, or it can be implemented as an archival tag.

It is preferred that the time stamping error is kept to within 2 seconds but "it will work with time errors of minutes if you receive signals from enough satellites". This comment leads the author to assume some further processing can be implemented using an over-determined solution to estimate the actual time error.

In terms of performance, the TTFF is quoted as being <100ms and the accuracy is 55 metres (95%). A sensitivity figure of 34dBHz is close to the performance of the SiRF chipset.

A figure of 3 Joules/Fix for power consumption has been provided through direct correspondence with the designer of the Fastloc system. This is significantly better than SiRF, mainly due to reduced ON time requirement.

Other key factors are;

1. Small size.
2. Easily integrated to communications link. Due to the limited amount of data required to be sent (code-phases, PRNs and time) it can even use ARGOS' limited bandwidth.

2.2.6 TIDGET™ Technology

TIDGET was developed and patented by Navsys Corporation in 1992 [52]. It was designed to be a low-cost GPS sensor for military applications. The sensor approach relied on a communications link between the sensor itself and a central processing station. The goal was to have a sensor that was expendable while utilizing a low-bandwidth communications channel to transfer the signal to a processing machine capable of processing the data.

At that time there were options for customer to use a conventional GPS receiver for tracking or a GPS "translator". The former was a fairly expensive option that was also physically large, while the later although very small and compact was very expensive.

TIDGET filled a niche in the market for expendable GPS sensor. It successfully found its way into application involving tracking Weather Balloons [53] and tracking missiles

[54]. Both applications assumed the sensor to be expendable and therefore cost was a critical factor. The missile tracking application for TIDGET also benefited from the extremely fast signal acquisition time achievable using the TIDGET architecture. As conventional GPS could not compete with the TTFF (Time-To-First-Fix) performance of a TIDGET system especially as the sensor would be stored with no visibility of any satellites for many months before being launched on the side of a missile with its extreme dynamics.

The TIDGET technology differed from the subsequent creation of TrackTag in that TrackTag stores the RF spectrum rather than buffering and sending over a telemetry link. The concept, however, remains very similar in terms of what processing is required to be done and TrackTag is considered an extension of the TIDGET technology.

TrackTag is based on the TIDGET technology patented by NAVSYS [55]. The aim of TrackTag is to offer the market a very small tag that can tell the user where it has been – anywhere on the planet.

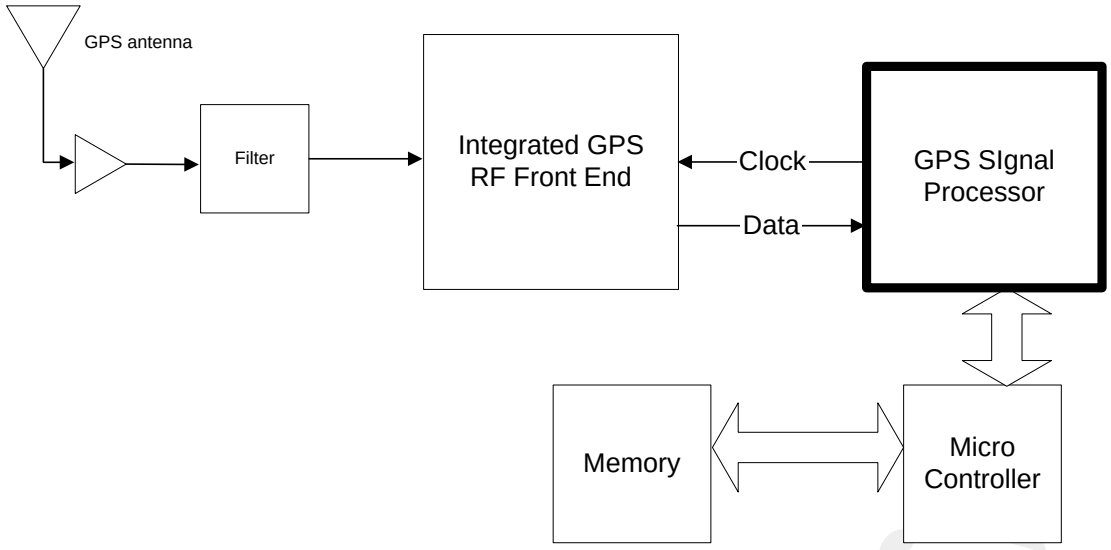
In applying the TIDGET technology to applications, such as animal tracking, there are major differences in how the design is approached;

1. Real-time processing is no longer required. Post-processing is suitable.
2. Signal Acquisition is potentially far more difficult as the previous TIDGET applications had a good view of the sky.
3. Miniaturisation is vital in order to be small enough for as many animals as possible.

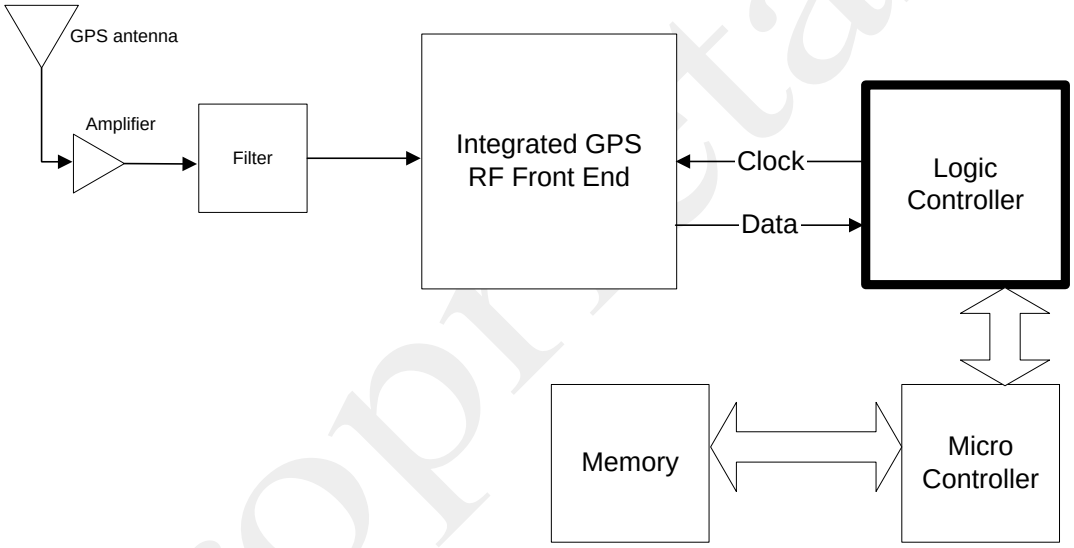
It does this by sampling the RF spectrum at the GPS L1 frequency and storing those samples in memory without any processing. The samples are downloaded into a file on PC and can then be processed by NAVSYS to find out where the tag was. The fundamental difference between this technology and other GPS tags is that no signal processing is done on the tag itself. This post-processing technique allows for an extremely small size and low-power design which should be a major benefit to some applications.

TrackTag has been designed to overcome the limitations of standard GPS receivers, which cannot be run for long periods of time using small batteries. A conventional GPS receiver, if woken up every hour, will (optimistically) need powered for at least 6 seconds to compute a navigation fix. By contrast, TrackTag needs to be powered up for around 20 milliseconds to acquire enough data for a GPS fix. This power saving is achieved by the use of the NAVSYS patented TIDGET process, whereby the raw GPS data is acquired unprocessed, and the time-consuming and power-hungry GPS signal processing is deferred until the data is retrieved from the TrackTag.

The major difference in the receiver architecture is the replacement of the conventional GPS processor with a small low-power logic controller in the TrackTag design shown in Figure 6. This is possible due to relatively simple process TrackTag is required to do, i.e. only saving the sampled data to a memory storage rather that undertaking computationally intensive GPS tracking algorithms.



A Conventional GPS receiver



B TrackTag Architecture

Figure 6. Receiver Architectures

The TrackTag system design differs from the conventional approach to GPS receiver design in that it defers all the computationally intensive DSP calculations until after the tag is retrieved. The TrackTag design simply records the received baseband signal to a large on-board Flash storage device. The Doppler removal, de-modulation and navigation calculations are all post-processed on a PC.

The ability to use very short snapshots could allow the tag to be made much smaller and therefore reduce overall weight, primarily by reducing the required battery size. The short snapshots could also offer more robust performance in harsh environments where

a long un-interrupted period of signal reception is not likely, such as in the marine environment.

The TrackTag is programmed to wake up at a user-definable interval (seconds to hours) to take a position fix. At the time of each position fix, an auxiliary data set is also automatically logged. This auxiliary data comprises: Tag temperature, an uncommitted analogue value, and two uncommitted digital values (an external connector permits the input of these uncommitted signals by the customer). The Tag can also optionally be woken by an external event (for example a surface-break detector if used for marine mammal tracking).

More details of how the TrackTag system was designed to form the platform for this research can be found in Appendix B: Platform Development.

2.3 TIDGET in Animal Tracking (TrackTag)

2.3.1 Performance

The ability to compute a position fix with very little data (20ms) means that the tag should be more robust in harsh environments. The patented design [55] could allow tracking where other GPS systems struggle to acquire the signals required, such as under thick forest canopy or in a marine environment.

2.3.2 Duration

Ultra-low power design, coupled with ability to use very short signal snapshots, leads to the tag being able to have a longer potential period of use. This can be achieved while maintaining the same position fix interval (duty cycle).

In comparing against competing GPS technologies, the SiRF™ [56] technology chipset is used as it is currently leading-edge conventional, low-power GPS design. Figure 7 shows the relative energy required by TrackTag, SiRF and Fastloc for a single position fix. At 20mJ TrackTag offers an order of magnitude lower power consumption than the example SiRF chipset (9.65J) and the Fastloc device (3.0J). This is due to the ON time requirement being so much lower – the instantaneous current drawn by TrackTag is quite high (100mA compared to SiRF's 65mA) but the time it is on is measured in milliseconds rather than seconds.

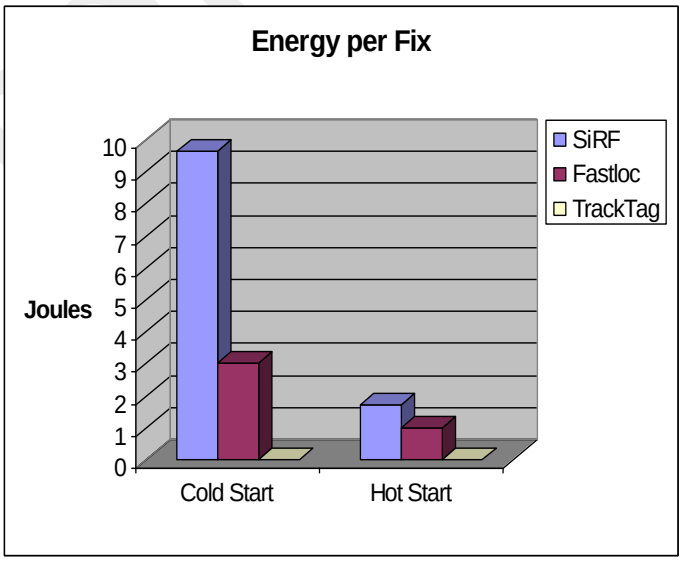


Figure 7. Comparison of Energy Per Fix

Figure 8 shows the number of possible fixes assuming a 3 gram (120mAh Lithium Polymer) battery is used. The estimated number of TrackTag position fixes would be 72,000 compared to SiRF's 830 or 147 depending on whether the fixes are all "Hot Starts" or "Cold Starts" respectively. In reality, the SiRF chipset will require an increasing number of "cold starts" for longer fix intervals but it is difficult to quantify the percentage split between start modes and it would depend on the application environment. TrackTag does not suffer from longer "cold start" time requirement but is able to take far more position fixes even when compared against the SiRF with all fixes "Hot Starts". Latest chipsets, such as SiRF, also support "trickle power" which switches the receiver off for a number of seconds between position fixes. This can reduce the average power consumption significantly but will still be well above that of TrackTag.

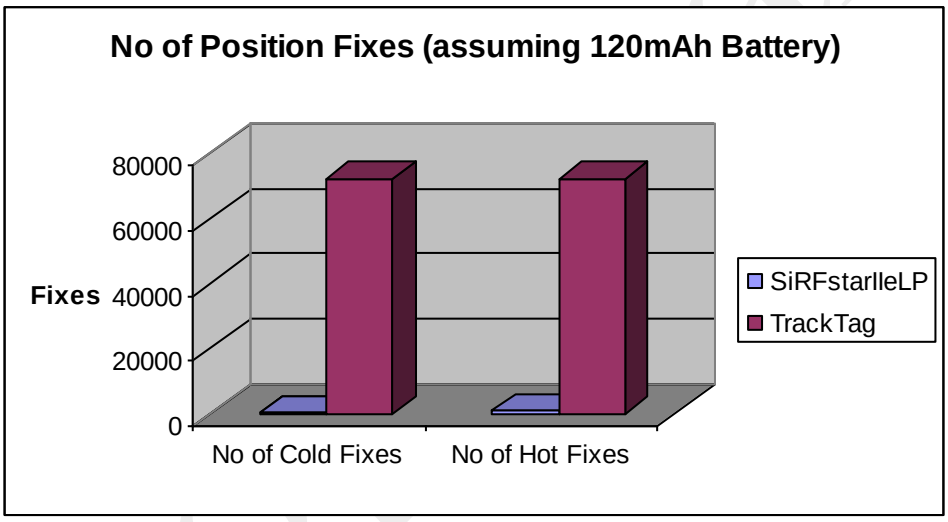


Figure 8. Comparison of Maximum Number of Fixes

NOTE:

Hot Fix:- Receiver powering up with current data (Position & Ephemeris).

Cold Fix:- Receiver powering up with Almanac data.

2.3.3 Size

"If a transmitter is less than 4% of the body weight, the transmitter does not affect the bird" [8]. This implies that any wildlife tracking study that uses a device that weighs more than 4% of the animal's weight could offer non-representative results due to the tag itself influencing the creature's actions.

This should be a critical factor when biologists are assessing which technology to use in their studies. If they use a tag deemed too heavy by others in their field of research, their results can be discredited.

The performance and duration strengths of TrackTag combine to allow for smaller batteries to be used in TrackTag for an equivalent number of position fixes. After all, as Figure 8 illustrated, so many more fixes are possible with TrackTag over a conventional GPS logger so a reduction in the tag size through using a smaller battery is possible.

One of the most common activities for animal researchers is the tracking of pigeons. With pigeons weighing between 500 and 600g the tag would have to measure less than 20g in total to satisfy the 4% rule. The smallest GPS tag available, at the time of writing, was a 30g GPS/PTT tag from "Microwave Telemetry" [57] which should only be used with animals as small as 750g and had a very limited possible number of fixes. There was therefore a definite need in the animal tracking community for a smaller device that could offer GPS positions in a smaller package. With the minimum weight of the proposed TrackTag design being 13g, there is still room for manoeuvre regarding the choice of battery, antenna or packaging while offering biologists the opportunity for extending their research capabilities.

2.3.4 Market Positioning

The application of TIDGET technology, in a package tailored to suit the animal-tracking application is known as Tracktag™. Based on the S.W.O.T. Analysis, conventional GPS tags are considered the main threat to TrackTag technology as they have the closest match in terms of capabilities. TrackTag is therefore now compared against conventional GPS tags in depth.

Based on a Market Segmentation Appraisal [7] there are two main areas open to TrackTag at present. They are animal tracking and surveillance with the former showing the largest potential.

Table 2 shows some global market data was provided by a large GPS & Radio-Telemetry tag vendor. The TrackTag estimates are added.

	GPS	VHF	ARGOS	TrackTag
Date first used	1994	1969	1987	2004
Animals that can be tagged	> 10kg	Any animal	>250g	>325g
Length of time tag deployed	Up to 2 yrs	Up to 2 yrs	Up to 3 yrs	Up to 4 yrs
Accuracy	3 - 10m	50 - 150m	100 - 4000m	100m
Potential No. of locations	Limitless	500	1000	64,000
Current cost per unit	£1500-£3000	£70 - £400	£1000 -£1500	£1,600
Recurring battery costs	£150-£250	£40-£100	N/A	N/A
Data cost/Staff costs per location	Free/£1	Free/£60	£35/£1	£0.1/£1
Current market share	250	60,000	1,000	10

Table 2: Wildlife Tracking Market Split by Technology

The company also stated:- "…the GPS market is between 300-500/year but the market for the normal VHF tag is about 100,000 per year – i.e. Biotrack in the UK sell 8000 VHF tags each year and they are small when you look at the USA market."

It is assumed that the table provided was a year or so out of date based on the market size being slightly less than their current estimate. So, why is the GPS tag market so much smaller than the VHF radio-telemetry market? The major reason is likely to be weight.

2.3.5 Early Results

In early 2004, during the 1st few weeks of this research, data was collected from an albatross that had been tagged using the TrackTag system in the South Atlantic. The flight lasted three weeks and the TrackTag data retrieved offered the user significantly increased accuracy and resolution to what they had been used to.

The flight path (Figure 9) follows a near figure-of-eight starting and finishing on South Georgia in the South Atlantic. The bathymetry (sea depth) data was included to show the map of the sub-sea geography. Day and night position fixes have been indicated by colour – white for a daytime fix, black for a fix during the night. Around 2500 fixes were computed equating to a fix around every 12 minutes on average. The number of positions recorded coupled with their accuracy offered the user more insight into how the bird navigates.

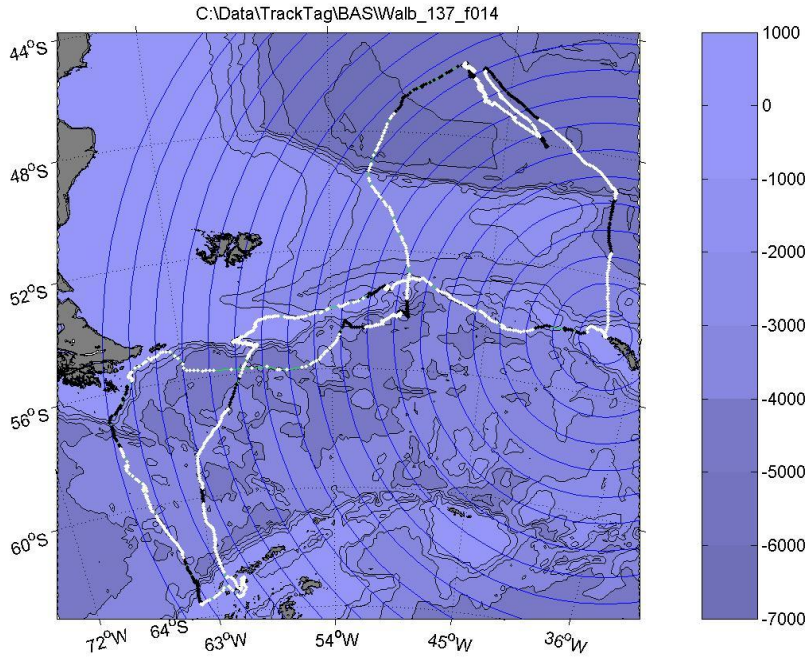


Figure 9. Flight of Albatross in South Atlantic

TrackTag shows potential with a key advantage over the competitive offerings in terms of power consumption. This is vital to compete in terms of a long-duration tag. However, there has been no work on the time correction issue. Although the 3-week trail (on an Albatross) was successful, it was close to the limit in terms of tag clock drift as it can be observed that the number of successful positions fell dramatically during the time where the clock error was greatest – around the middle of the data as the start and end time was known. For TrackTag to operate over long durations (>1 year) a new approach is essential to enable the correction for clock drift. This is the key reason for the research described in Chapter 7: Time Correction.

2.4 Focus for TrackTag Research

The primary advantages of ultra-low-power operation and the ability to operate in harsh environments are a result of the unique architecture employed by TrackTag. The TrackTag architecture has been demonstrated to work using only around 20 milliseconds of data and this is what enables the system to operate using such little power and also makes it more robust due to there being less dependency on the length of uninterrupted GPS signals received.

To turn the TrackTag concept into a viable product there are a number of challenges that require further research. There was a large part of work required to develop the product to enable the research and this was described in Appendix B. The next five chapters, however, are arranged in such a way to deal with the challenges in the context of the PhD research. Those chapters have the following order;

3. Antenna Integration. Closely linked to the "Weak Signal Detection" work as it is basically a prerequisite for doing any weak signal processing. How the antenna was selected as well as how it was optimised depended on the test setup employed. It was this testing apparatus that was novel.

4. Signal Detection. The building blocks for basic TrackTag signal processing are described here. Algorithms that take raw data and provide navigation solutions for real-world data are developed. These algorithms highlight the limitations of conventional approaches to the TrackTag signal-processing challenge.

5. Weak Signal Detection. The system, as it was before any development through this research, used a very simple signal detection process which is not capable of detecting weak signals. It was understood that if the tag was to be used in real-world animal tracking, weak signal detection would be key to enabling the device to work in many animal habitats. The requirement for weak signal detection would ordinarily push the design of any receiver toward taking longer sets of data in order to allow more processing. However, in this research, there is a desire to maintain the very short snapshot length of only around 20 milliseconds to maintain TrackTag's ultra-low-power consumption.

6. Signal Detection Enhancements. Some improvements are described here that improved the performance of the TrackTag system in general.

7. Time Correction Algorithms. Due to the architecture of TrackTag the tag relies on an on-board real-time-clock to keep track of time. As an "ultra-low-power" tag will be of most use in very long duration studies, the ability to maintain accurate time estimation is crucial. Advanced methods for the correction of estimated time were developed during this research and are discussed.

The last chapter concludes by showing some animal tracks taken by biologists using TrackTag during the development of this research.

Chapter 3: Antenna Integration

3.1 Overview

The most critical area of hardware development on the tag undertaken was the integration of a miniature antenna. With a miniature GPS tag the antenna performance is a major issue. Miniaturisation of the antenna sub-system, while maintaining sufficient gain is a key parameter to enabling the technology. Although there were a number of antenna options, none had been demonstrated to work reliably.

Antenna integration for miniature GPS devices was a major challenge to the development that consumed many months of effort. A major problem was the lack of a consistent method for testing the antenna and RF front-end performance. With wildly variable results it was difficult to ascertain what sub-system was at fault in many cases.

The variables in question were;

1. New PCB layout
2. New, miniature antenna
3. Different batch of RF parts
4. Antenna Orientation
5. Antenna Near Field environment
6. GPS satellite positions
7. Tracking algorithms not fully developed

With hindsight, it was discovered that all of these variables, except (1), were influencing the tests being carried out at the time that the tag RF front-end and integration of miniature antenna were under development. In order to make progress a novel approach to aid antenna integration was therefore required that would reduce the variables involved in the test.

An innovative "Turntable Test" was developed that enabled something close to a full antenna gain pattern test using a simple turntable setup and relying on the GPS satellite motion to collect all angles required for a 3D gain plot. This would result in tests being carried out where the antenna parameters could be controlled and GPS satellite positions

would also be taken into account. All other variables could be removed through part-to-part comparison while keeping the algorithm constant.

The assumption was that a 12mm "Patch" antenna would be used in the standard TrackTag design. These tests, however, were to conclude that mounting a patch antenna on the TrackTag board itself gave very poor performance. This was subsequently put down to diffraction of the antenna field pattern around tag's ground plane. Diffraction of the antenna's gain pattern around the tag's ground plane occurred due to the ground plane being far too small. A potential way to resolve the diffraction issue was to increase to area of the tag ground plane but it was found that the required ground plane would make the tag as a whole far too large and so was not a practical option. The effects of the diffraction were made even worse when the tag was mounted on an animal because the antenna's near-field encountered tissue proximity effects which significantly attenuated any signal.

The turntable test (described in the following text) provided a method for experimenting with some antenna types and orientations together with "simulated tissue" in various positions. The "simulated tissue" was created by soaking a cloth in salty water and then wrapped in a sealed polythene bag. When placed next to the tag, it caused similar performance degradation as various meats that had all been shown to affect the signal reception adversely. The salinity was not measured. This was used in conjunction with NAVSYS' packaging expert in order to design appropriate packaging for the species the end user wanted to study.

3.2 Turntable Concept

The idea of conducting anechoic tests outdoors is not new. There has been some work done in the field of EMC testing outdoors for equipment that is too large to be accommodated within a chamber [58]. In those outdoor tests care is taken to characterise the environment in which the tests are carried out. The major difference between an outdoor test and a real anechoic test is the lack of screening. Having no screening introduces two sources of error, namely reflections/multipath and EMI (Electromagnetic Interference). These errors can be minimised through characterisation of the test site before testing is undertaken.

This technique uses the same signals that the system will use in operation. This offers data that is truly representative of how the system will behave in the real-world scenario it is being designed for. No characterisation is performed and therefore the results are an indicator of how well the complete system will perform in that environment.

Assuming the test is conducted in the same place, different design options (usually different antennas) can be compared in terms of their gain pattern using the turntable technique. Observations have shown that the satellite's transmission power, although very stable, does vary with elevation. This is due to the satellite's antenna beam pattern and must be taken into account.

Figure 10 shows the basic concept behind how the turntable rotation is coupled with the satellite motion in order to present the DUT (Device Under Test) with as many angles as possible. Depending on the constellation geometry at the user's location, many satellites will pass overhead. The test will give the best coverage when a satellite is used which covers the entire range of elevation angles from 0° to 90°.

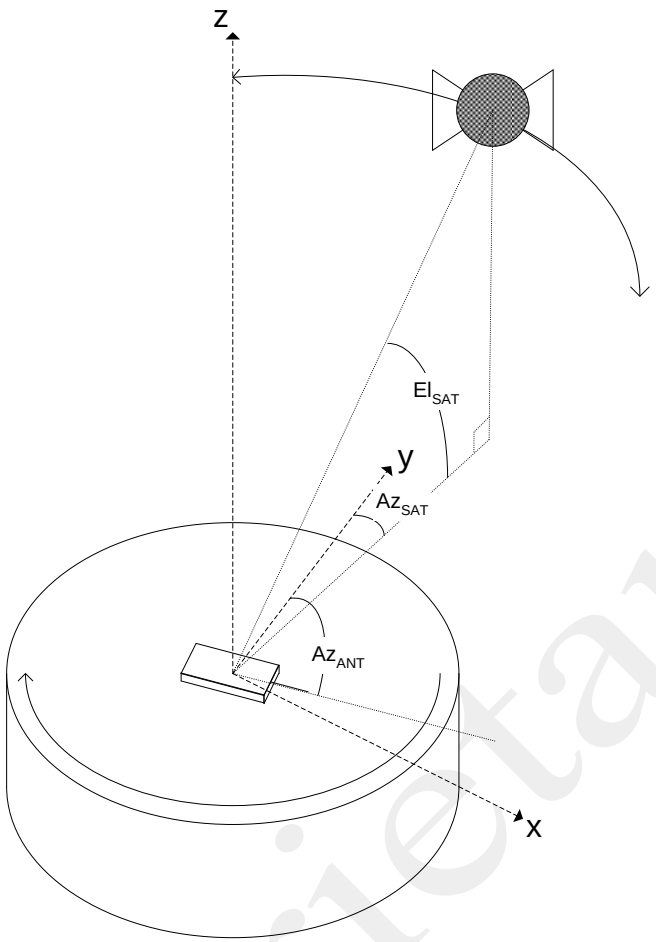


Figure 10. Turntable Hardware Setup showing Tag to Satellite angles (Az_{ANT}, Az_{SAT} and El_{SAT})

The turntable uses a simple DC motor to rotate. Each step is controlled to be exactly 18°, therefore requires 20 steps for a complete rotation. Each turn's timing has to be precisely controlled and is done by synchronising the control of the DC motor to GPS time by using the 1pps (pulse-per-second) generated by a stand-alone GPS receiver to start the step. An optical sensor on one on the gears is then used to sense when to stop the turn.

The DUT itself has to be able to either record the received signal data or send it to a system that can. There are battery powered GPS units with a Bluetooth link which could be used and avoids the need for slip rings to handle a wired system on the turntable. At NAVSYS the system was used to characterise miniature GPS data loggers which are therefore designed with such capability. DUTs that cannot store the signal data (if an antenna itself is under test for example) can be coupled to commercially available GPS logger. The logger should record the following data at every turn; Time,

All tracked satellite SNRs (Signal-to-Noise Ratio) and all satellite azimuth and elevation angles. This data can be found in standard NMEA format message often provided by GPS devices.

Software is available on the market, such as the NAVSYS Matlab toolbox (Figure 11), that will predict which GPS satellites will be in view and at what angle. This software is used to help plan the best period of time to run the turntable test and acquire data for a good range of elevations.

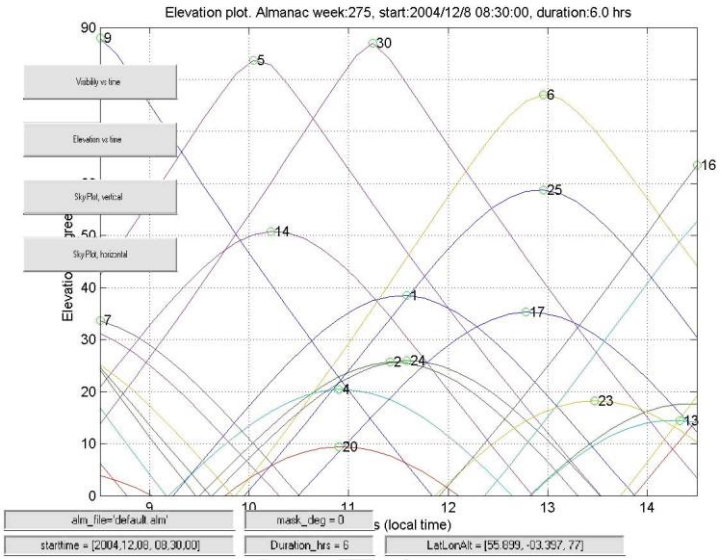


Figure 11. Satellite Elevation Prediction

Figure 12 shows the calculated angles, taken from real data, on an "Azel plot". The way to interpret the plot is to imagine any points on the circumference are on the horizon (0° elevation). Any points in the centre of the plot correspond to measurements taken while the satellite is directly overhead (90° elevation). As the turntable rotates, the Azimuth angle with respect to the DUT is changing. The DUT azimuth angle (Az) is therefore calculated using equation (7).

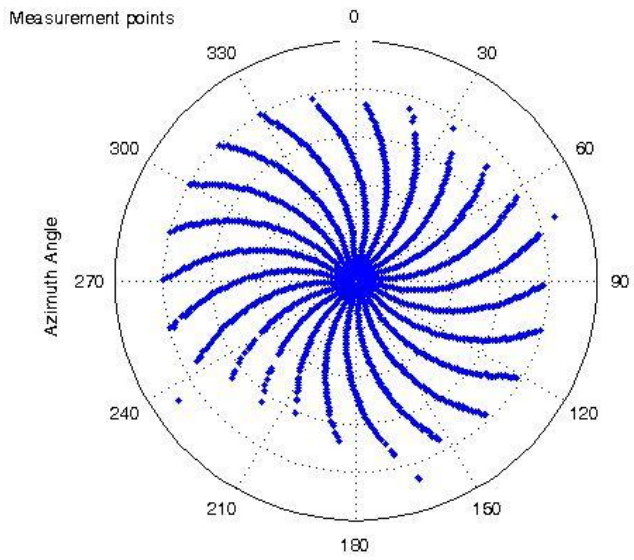


Figure 12. Typical Turntable Measurement Points

$$Az = Az_{ANT} - Az_{SAT} \tag{7}$$

The turntable/antenna angle, Az_{ANT}, is derived from the GPS timestamp of the measurement and working out how many step rotations have taken place. The satellite azimuth angle, Az_{SAT}, will be provided by the logger's GPS processing or can be calculated by post-processing using the GPS time reference and archived GPS orbital Ephemeris data. The elevation angle is provided by the logger's internal GPS processing (or can also be derived using the GPS timestamp and Ephemeris) and assumes that the turntable has been setup level.

Figure 12 also highlights a limitation of the turntable approach. There are no points provided by tracked signals on, or within around 20° of the horizon. The test is limited by the site surroundings which can prevent signals being acquired at low elevations. The data shown here was taken where there were some trees and bushes encircling the turntable, hence the need to do any device comparison tests at exactly the same site. This limitation, however, offers an insight into how well the device will work in that environment and represents the real-world antenna performance there rather than a near-perfect but unrealistic lab environment.

Having recorded the data for the angles shown in Figure 12, a correction is required due to the variance in power received from the satellite with respect to the elevation angle. This is due to the transmission loss not being equal over all elevation angles. The attenuation, Δ_{SAT}, due to the satellite gain pattern can be estimated using equation (8)[48], where E is the satellite elevation and k = 2.5413.

$$\Delta_{SAT} = k(1-\sin E) \tag{8}$$

This indicates that at 10° elevation there will be 2.1dB attenuation due to the satellite antenna's gain pattern. This therefore has to be accounted for in the turntable test. The Satellite's antenna gain is assumed to be uniform over its main lobe which covers the Earth.

Atmospheric attenuation in the frequency band of interest (1-2GHz) is dominated by oxygen attenuation. This can be approximated by equation (9)[48], A_{oxy}, where $a = h_m/R_e << 1$ and h_m is the equivalent height for oxygen, 6km, and R_e is the Earth's radius (6378km).

$$A_{oxy} = \frac{2\Delta_{90°}(1+a/2)}{\sin E + \sqrt{\sin^2 E + 2a + a^2}} \tag{9}$$

This effect equates to around 0.2dB attenuation at 10° elevation and less at elevations above. As there is little interest in the gain pattern at elevations below 10° it was decided that this effect would be ignored.

The SNR measurements for all points taken from the satellite showing the highest elevation angle range are adjusted for the satellite gain pattern using equation (8).

3.3 Turntable Results

The data collected takes the form of a sparse matrix as there are an infinite number of possible angles as illustrated in Figure 12. Three dimensional interpolations are therefore utilised to create a surface representing the gain.

It is important to note that it is actually Carrier-to-Noise Density (C/N_0) represented in these plots and therefore includes the characteristics of the tag's RF Front-End Gain as well as the Spreading Gain because the signal detection de-modulates (de-spreads BPSK) the signal before estimating the C/N_0. In general, a signal offering > 40dB-Hz is considered strong, while a signal < 34dB-Hz is considered weak. What signal level is required is a major topic covered in the "Signal Detection" chapters.

Figure 13 shows the antenna gain pattern seen while using a simple ¼ wave, monopole, whip antenna. As the device was placed on the turntable horizontally there are areas on the horizon where the gain is low. These correspond to the direction the whip is pointing. Higher gain is observed at a 90° shift in Azimuth angle. This suggests we have the expected "doughnut" gain pattern for the whip antenna.

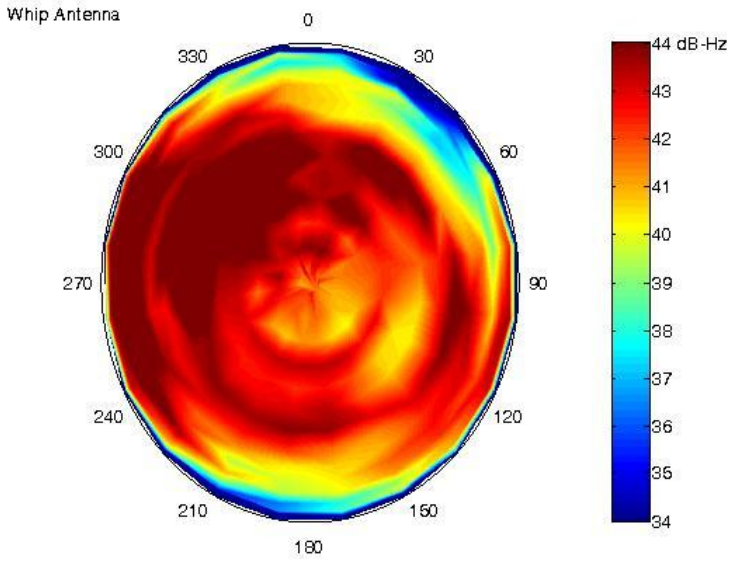


Figure 13. 'Azel' Gain Plot for Whip Antenna

The same measurement data is used to produce the standard gain plots shown in Figure 14 and Figure 15.

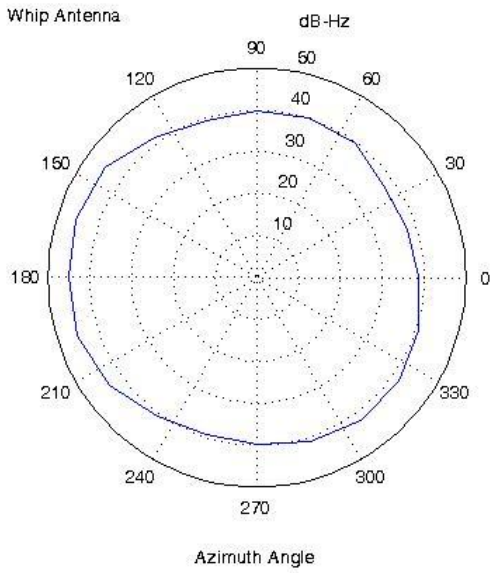


Figure 14. Azimuth Gain Plot for Whip Antenna

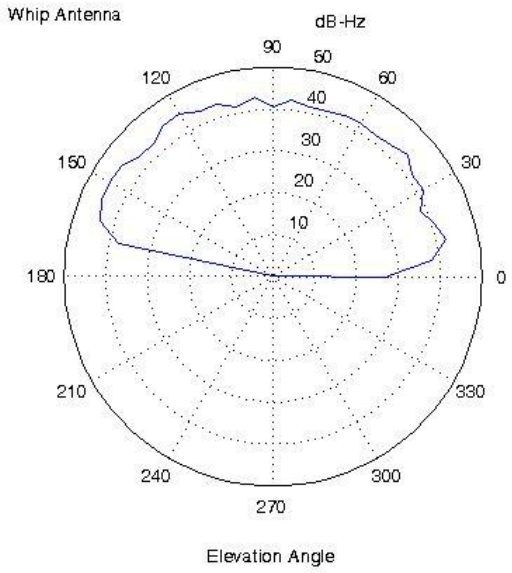


Figure 15. Elevation Gain Plot for Whip Antenna

Figure 16 shows the result provided by the turntable test when using a ceramic chip antenna (Phycomp). The Phycomp had previously been so close in performance to a Whip antenna that it was uncertain which one offered superior performance. These results indicate that the Ceramic Chip antenna gives a more uniform gain pattern than

that of the Whip antenna. For this reason, the tag board was re-designed to take the Ceramic Chip antenna.

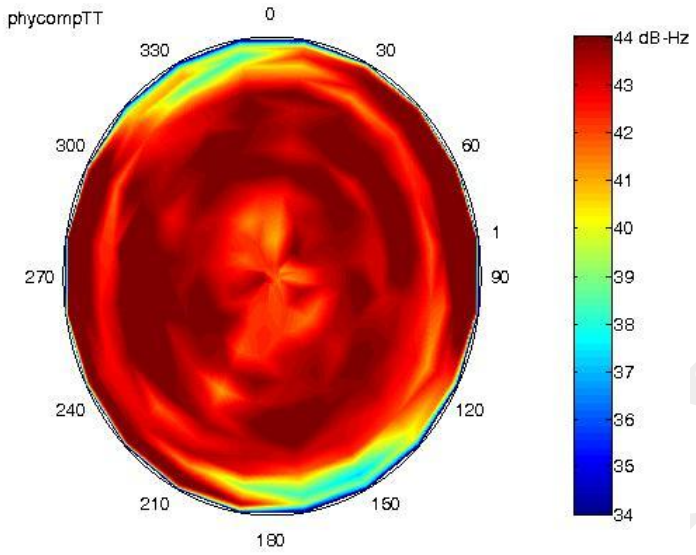


Figure 16. 'Azel' Gain Plot for Ceramic Chip Antenna

Again, the conventional gain plots are shown for the Ceramic Chip antenna in Figure 17 and Figure 18.

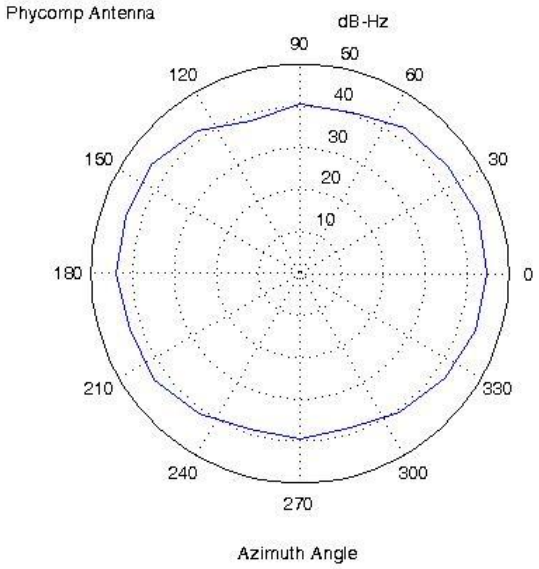


Figure 17. Azimuth Gain Plot for Ceramic Chip Antenna

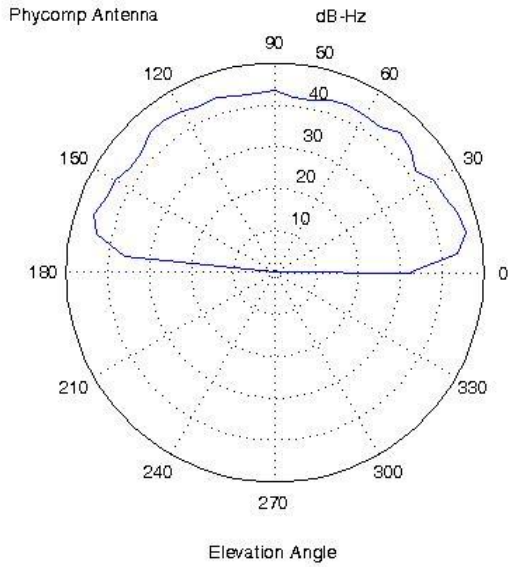


Figure 18. Elevation Gain Plot for Ceramic Chip Antenna

3.4 Summary

The tag hardware was a major undertaking due to the antenna integration being problematic. The problems encountered were mostly due to the fact there were so many variables all playing a role, some of which were not even known from the start – tissue proximity in particular.

There was re-layout on the board required, mainly to accommodate different antenna types but also to improve its "Design for Manufacture" which involved making the design easier to build and more robust. This not only resulted in reducing the amount of time required to build units and have them pass performance tests, but there was also more consistency in performance of units.

A major re-appraisal of antenna part selection resulted as the research concluded that the assumed antenna of choice (12mm patch) would not work on such a small tag. This decision was not taken lightly as many antennas had already been purchased. This work also resulted in the development of a very useful piece of test apparatus that should prove invaluable to NAVSYS in the integration of future antennas and/or miniaturisation of the tag in general.

Chapter 4: Signal Detection

Detection of the satellite signals is difficult to achieve in harsh environments and/or with limited antenna performance. The signals transmitted by the GPS satellites utilise the benefits of spread spectrum communication in order to allow these signals to be detected on Earth even though the signal power received is below noise. With such a limited power budget the issue of accurately de-modulating the signal is not straight-forward.

This chapter focuses on the challenge of detecting GPS signals while still limiting the length of data used to around 20 milliseconds.

The basics of GPS signal structure and receiver design have already been discussed in Chapter 2. Specific detail of how the signals were detected during this research will now be described and will then follow on to discuss how they were developed into more advanced algorithms capable of weak signal detection within the constraints of the TrackTag system design.

4.1 Tracking Correlator Technique

The conventional correlator approach is used in many GPS applications. The design of a correlator can be made relatively simple and was therefore the 1st step in this research. The TIDGET™ concept had already been proven using software implementation of tracking correlators and so it was already known that the technique would work.

It was not intended that the correlator would be used to detect weak signals as the technique is prone to false locking. It was anticipated that the "FFT technique", discussed later, would perform far better in terms of false-locking. In many of TrackTag's applications, weak GPS signal processing is required and setting the correct thresholds in a correlator to prevent false locking would be a major problem. The correlator was therefore implemented to provide the baseline approach from which further techniques could be analysed.

Implementation of a tracking correlator would provide the following;

1. Doppler frequency measurement.
2. Code-phase measurement.
3. Carrier-phase.
4. Navigation Data bits de-coded.

4.1.1 1ms Correlator Engine

The fundamental building block developed for the correlator was the "1ms Correlator Engine". Shown in Figure 19, the concept at this abstract level is straight-forward. The incoming data is taken in 1ms chunks and multiplied against the predicted signal. As TrackTag's sample rate is nominally 5MHz, there are always 5,000 samples to multiply. The resulting 5,000 element complex vector is then summed and the magnitude is defined as the correlation value.

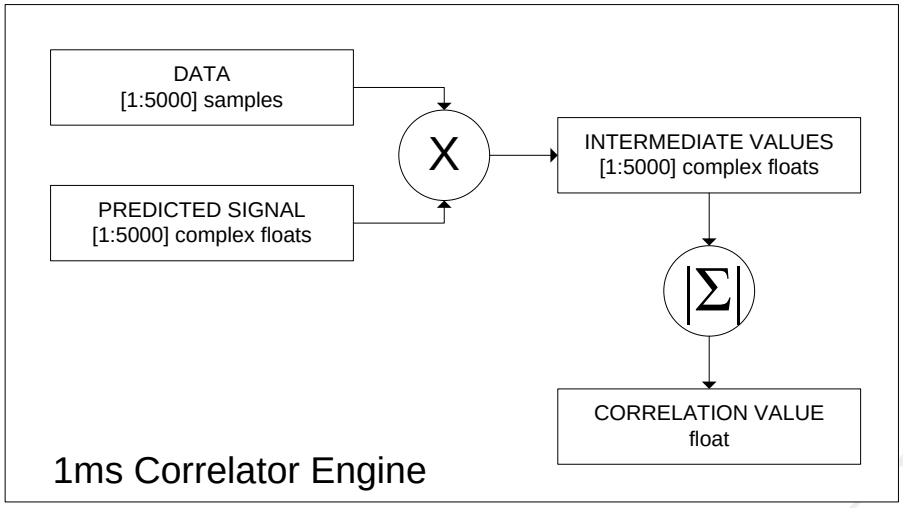


Figure 19. 1ms Correlator Engine

The actual correlation is therefore easily implemented in code. It is how to construct the input vectors that presents difficulty.

DATA

The DATA vector is the raw data taken from the tag's memory as a long vector of 2-bit samples. Around 131,000 samples are available, or just over 26ms worth, although this can include a period when the tag front-end oscillator was still stabilising. The snapshot length is set to 26ms purely to make the digital design straightforward as it equates to a snapshot size of exactly 32KB of Flash memory. Whichever millisecond was to be correlated would determine how far into the raw data vector to index in order to provide the 5,000 sample long sub-vector for correlation.

The millisecond indexing would also be offset by a certain amount so that the correlation boundary would coincide with any data bit transitions as it was found, later in the research, that if this was not done, one correlation could suffer reduced power due to being partially between navigation data bits and therefore the signal was inverted part of the way through the correlation. This could only be done if the code-phase for the signal of interest is known as every navigation data bit is synchronised with the BPSK code epoch.

PREDICTED SIGNAL

This simulates the down-converted GPS signal expected. The vector takes into account the carrier frequency, which is the nominal "L1" frequency offset by the Doppler encountered by the particular satellite at that instant in time. Of course, the carrier is

modulated by the BPSK spreading code unique to the satellite in question. However, even the BPSK code is stretched (or compressed) by the Doppler encountered and this has to be taken into account when constructing the expected signal.

So, the predicted signal is generated through use of the CA code generators and the Doppler frequency of interest. The code-phase of the signal must also be used in order to accurately reproduce the signal timing. Even though the code-phase is used to offset the DATA vector, that adjustment is relatively crude as it shifts the data index to the nearest sample and therefore has a resolution of around one fifth of a BPSK chip. A fractional code-phase (remainder once data offset is removed) is therefore required to maintain accuracy.

The last component to be adjusted is that of carrier phase. This is a simple offset for the carrier sine wave that is assumed. The predicted signal is generated as a complex floating point vector that corresponds to the incoming DATA vector of 1ms length. Complex notation is used to create real and imaginary components (I and Q) thus giving both signal magnitude and phase.

4.1.2 Tracking Loop Design

The overall design is shown in Figure 20. There are three tracking loops working together, one PLL (Phase-Locked-Loop) operating on the Carrier Phase estimate, one FLL (Frequency-Locked-Loop) operating on the Frequency estimate and a DLL (Delay-Locked-Loop) operating on the Code-Phase estimate.

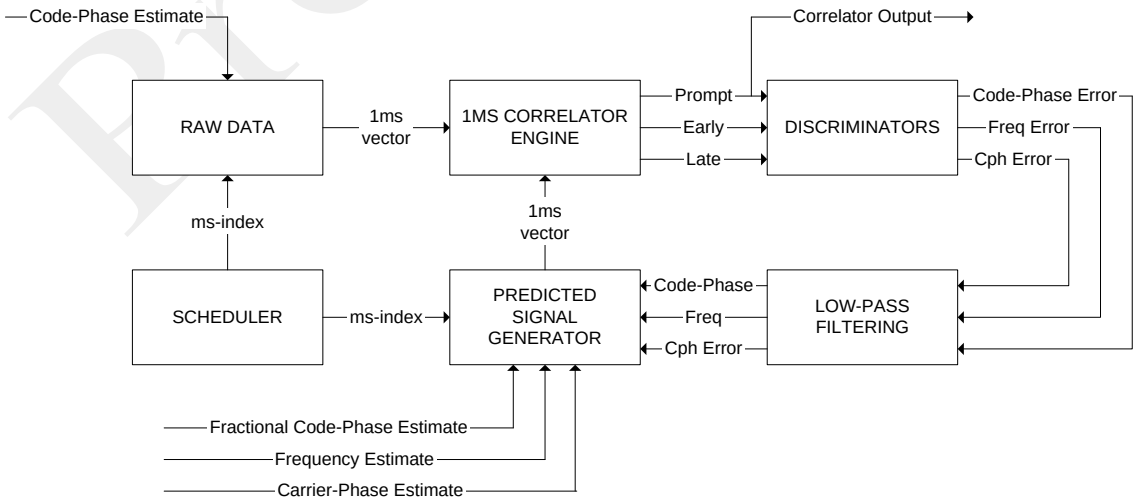


Figure 20. Tracking Correlator Design

As previously mentioned, an adjustment is made to the data pointer in order to align the data vector with the start of the BPSK code period. This assumes knowledge of the code-phase. It explains why the code-phase estimate is provided to the data vector input in such a way that the correlator system operates at close to zero code-phase and so only the "fractional code-phase estimate" is adjusted by the correlator.

The "1ms Correlator Engine" and "Predicted Signal Generator" are implemented together in C within Matlab in an effort to optimise for processing speed as they are processing intensive operations that are called many times.

Scheduler

In the TrackTag architecture the data length is very limited. For this reason, the correlator is forced to go through the data many times before the correlation settles. In an effort to prevent any sudden jumps in the signal components (i.e. potentially caused by jumping back to the start of the data once the end is reached) the correlator is scheduled to go backwards through the data after the end is reached. The scheduler will, in fact, repeatedly go back and forwards through the same data over and over until the desired correlation length is reached.

The direction of correlation has to therefore be taken into account when the propagation of tracking loop parameters is done.

Discriminators

There are various "textbook methods" to estimate the errors for each of the three tracking loops. The methods implemented are now discussed;

1. Delay-Locked-Loop

   The "Normalised-early-minus-late-envelope" discriminator was used [48, 59]. This requires a relatively high computational burden compared to simpler functions. However, this computation is considered negligible compared to that required for the correlation setup and execution and so discriminator complexity is not restricted. The code-phase error (11) is therefore estimated using the difference between early and late power measurements and normalised using their product.

$$P = \sqrt{(I+Q)*(I-Q)} \tag{10}$$

$$\tau_{err} = \frac{P_e - P_l}{P_e + P_l} \tag{11}$$

2. Frequency-Locked-Loop

   The "Cross-Product" discriminator (12) is used [48, 59]. Its output curve is proportional to the signal amplitude and is therefore has to be normalised using an estimate of the signal's SNR (Signal-to-Noise Ratio).

$$\begin{aligned} cross &= (Q_n * I_{n-1}) - (I_n * Q_{n-1}) \\ dot &= (I_n * I_{n-1}) + (Q_n * Q_{n-1}) \\ f_{err} &= sign(dot_P) * \frac{cross_P}{SNR} \end{aligned} \tag{12}$$

3. Phase-Locked-Loop

   The arctangent discriminator for Carrier-Phase measurement (13) is used as it is considered optimal [48, 59].

$$\theta_{err} = \tan^{-1}\left[\frac{Q_P}{I_P}\right] \tag{13}$$

Low-Pass Filtering

The discriminator outputs are used to adjust the current estimates for code-phase, frequency and carrier-phase. This adjustment is done by adding the estimated error term after it has been scaled by a damping factor. The damping factors are required to limit the bandwidth of the tracking loops and maintain stability.

4.1.3 Real Signal Detection

A snapshot of real data known to have a strong signal was used to analyse how the correlator performed. The correlator was set to run through the data for 30 cycles. This equated to the equivalent of 1350ms correlation time using 23ms of data.

The strength of the signal was estimated by simultaneous measurement on a separate conventional receiver. The signal strength was estimated to be just over 50dB-Hz on PRN-22. The Doppler frequency of the signal reported by the conventional receiver was also noted as well as its code-phase. These parameters were required as the correlator did not have either Doppler or code-phase search algorithms.

Code-Phase

Figure 21 shows how the correlator estimate for code-phase changed as its tracking loops pulled in. The code-phase shown is actually just the difference between the estimate provided by the "manual" measurement made on other system and the correlator measurement.

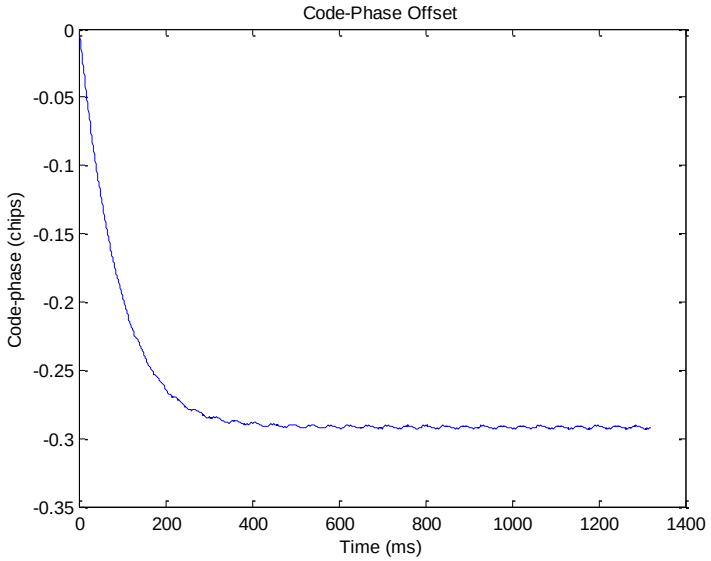


Figure 21. Correlator Tracking Code-Phase

The plot demonstrates that the correlator stabilises and indeed, although not shown, it stabilised on the same value when the initial code-phase estimate was varied by any amount within 1 chip of the final value. How much the initial code-phase estimate could be varied before problems arise is shown in Figure 22 where the initial code-phase is swept from -20 to +20 chips in ½ chip steps.

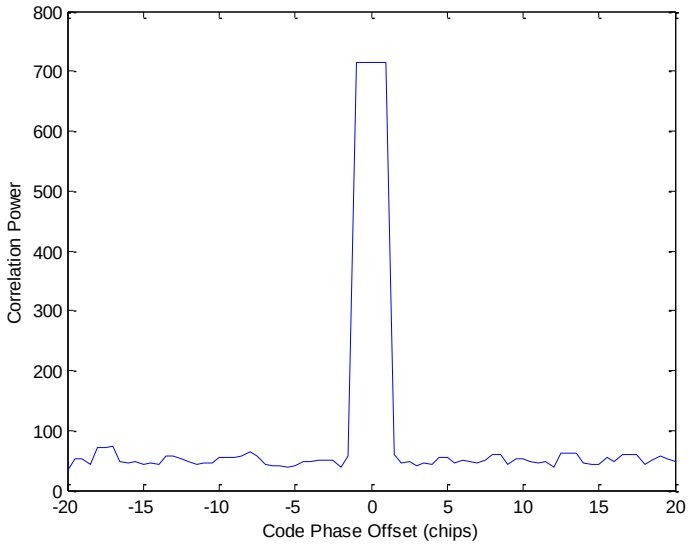


Figure 22. Tracking Correlator Power versus Code Phase Offset

It is clear that the correlator has to have an initial estimate of code-phase accurate to within 1 chip, anything more and the correlator does not stabilise. This is the expected performance of the DLL discriminator but does mean that if the code-phase is unknown every one of the 1023 potential code-phase phases would have to be searched in a code-phase search.

Doppler Offset

The estimated Doppler taken from the reference conventional receiver had to be added to the effective Doppler offset expected due to the drift on the tag's local RF front-end down-converter oscillator. The oscillator drift was measured manually then combined with the estimated actual Doppler frequency offset of the signal itself to give the initial Doppler offset for the correlator. This initial offset was then assumed as the reference zero point as the correlator locked on to what it estimated to be the actual Doppler frequency as the 30 cycles were processed. The result shows that the frequency locked loop also converged on a single value (Figure 23).

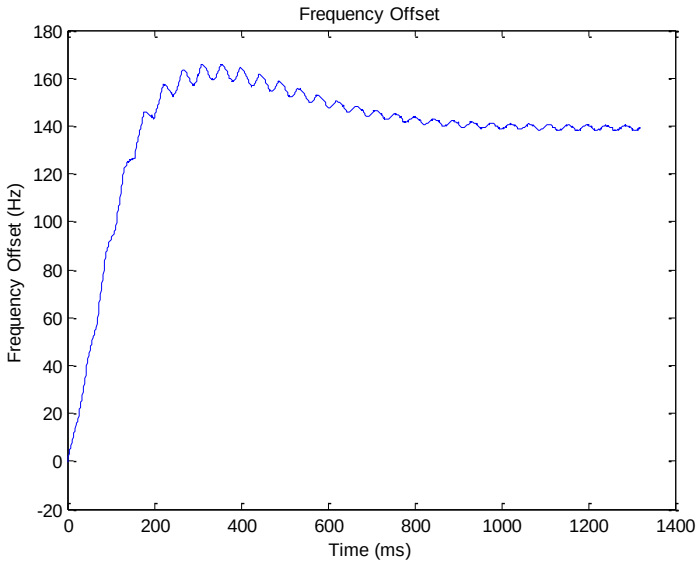


Figure 23. Correlator Tracking Doppler

As with the code-phase tracking loop, the frequency locked loop could be initialised using slightly different values for the frequency but always converged on the same value. How far this initial value could be was analysed by sweeping that initial frequency value. Figure 24 shows that when the initial estimate of the signal Doppler is within 250Hz the same correlation power is found, and indeed the same +140Hz result was obtained as Figure 25 illustrates.

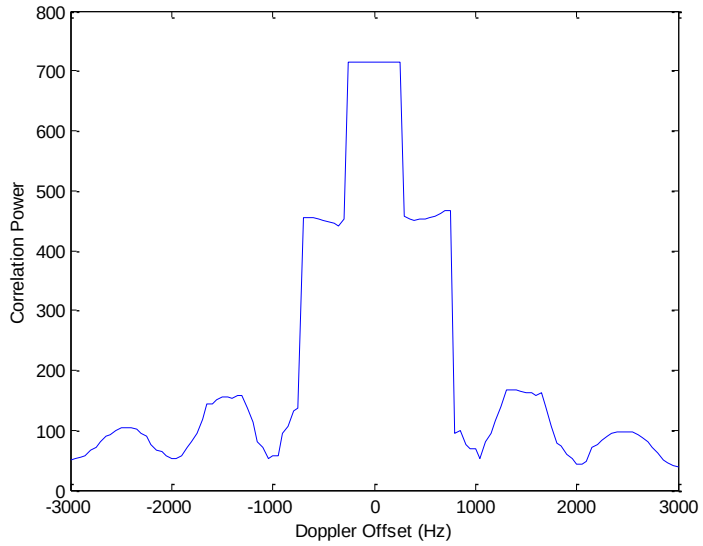


Figure 24. Tracking Correlator Power versus Doppler Offset

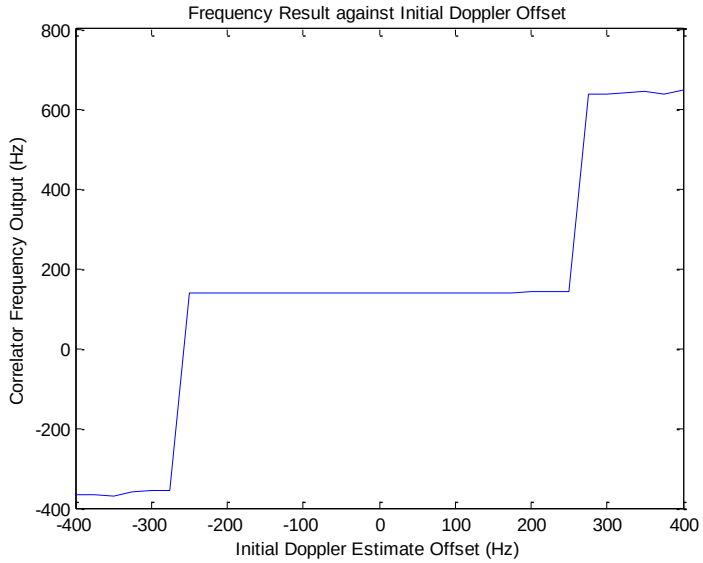


Figure 25. Correlator Frequency Measurement over swept initial Doppler estimate

Doppler and Code-Phase Search

The previous experiments using the Correlator to find a strong signal demonstrated that the estimates for Doppler and code-phase have to be within 250Hz and 1 chip respectively. Figure 26 graphically shows the problem of signal detection. The plot is limited to only 20 chips and 4kHz either side of the correlation peak. As it is shown here the peak is measured in terms of the correlation power. The next step was to quantify the signal power in real terms, i.e. dB-Hz.

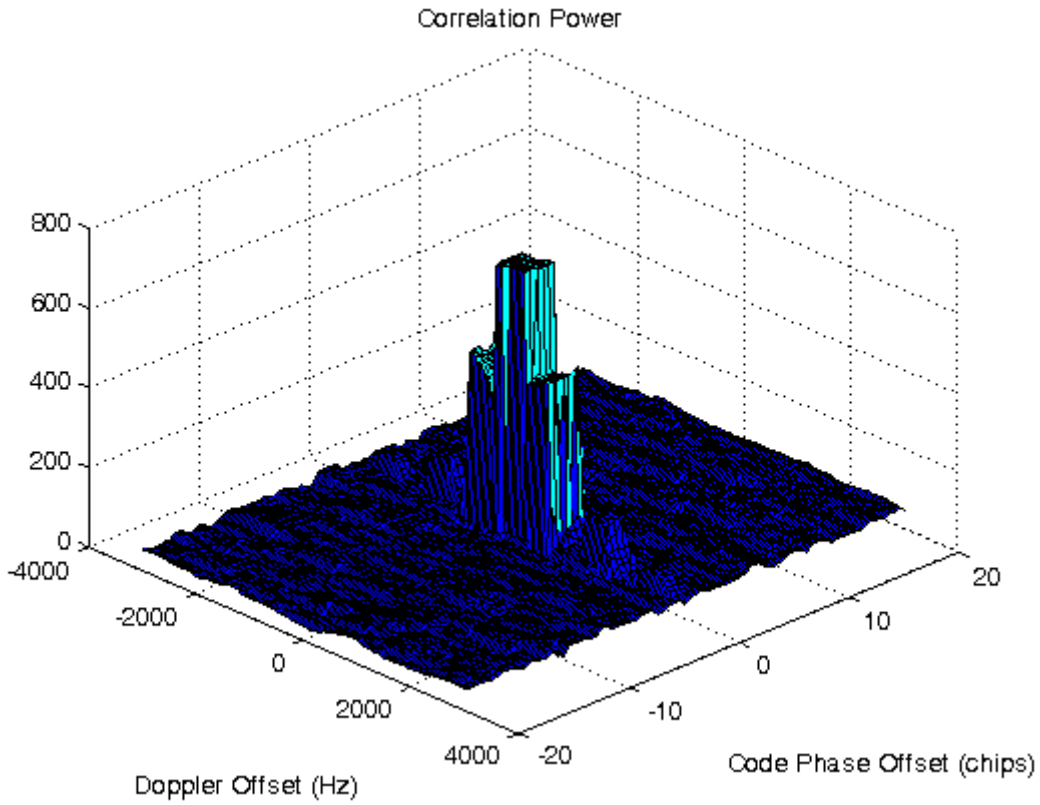


Figure 26. Tracking Correlator Power Surface

4.1.4 Signal Power

Setting the weighting for intermediate values

The expected noise level of the receiver is required before the signal detection levels are calculated. As the receiver noise floor is dominated by the front-end, two configurations were used to take data for analysis. One passive antenna and one active antenna were used in order to see whether different antenna configurations would affect the noise floor experienced by the receiver. All data was taken with the receiver's antenna obscured by a Faraday cage arrangement and also, well away from any potential EMI source.

The front-end chipset used was operated as a 2-bit sampler. This is done by use of the SIGN bit and the MAG (Magnitude) bit. The SIGN bit indicates polarity and the MAG bit is set when the magnitude reaches a certain value. This is controlled by an internal AGC (Automatic Gain Control) circuit and, according to the manufacturer's datasheet, should be set 30% of the time. Figure 27 shows the percentage of time that the MAG bit from the receiver front-end is set while using a passive antenna. Figure 28 shows the

corresponding plot for the same test but using an active antenna. These plots indicate that the front-end AGC is operating to the intended specification no-matter which antenna is used.

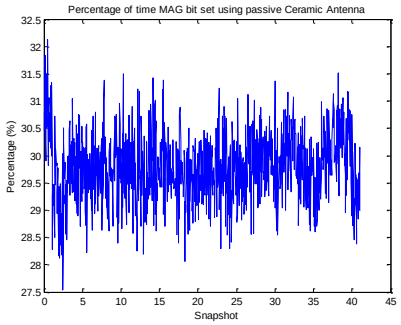


Figure 27. Percentage of time MAG bit set using Passive Antenna

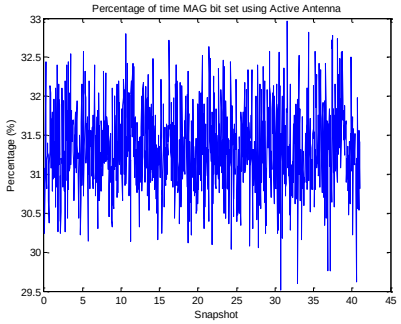


Figure 28. Percentage of time MAG bit set using Active Antenna

A numerical value had to be assigned to each of the 4 possible states that the 2-bit sampler can provide. This is effectively a weighting assigned to the intermediate values. It was set, arbitrarily, according to Table 3;

SIGN	MAG	VALUE
0	1	-1
0	0	-0.4
1	0	0.4
1	1	1

Table 3: SIGN & MAG bit to value

Given that the percentage of time the AGC will set the MAG bit is known and the

intermediate values are also known, the mean noise of the raw data can be predicted using equation (14);

$$E(n) = P_H + W(1 - P_H) \tag{14}$$

Where;

E(n): Expected Mean Noise Level

P_H: Probability of MAG being set

W: Weighting coefficient

In the active antenna case given previously, the percentage of time the MAG bit was set was slightly higher than the expected 30% and results in an average of a little more than the expected 0.58 (Figure 29). For the purposes of all following development, the MAG bit will be assumed to be set 30% of the time.

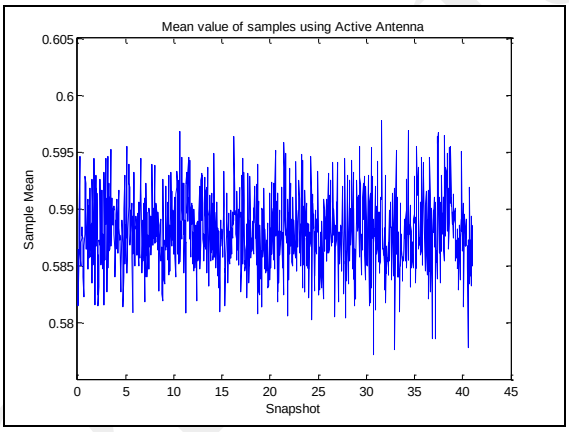


Figure 29. Mean Sample Value of Raw Data using Active Antenna

Correlation Output Noise

The noise level on the output of the de-spreading function was of more interest because it is this value that will be of relevance when deciding whether a signal is present or not. The noise floor for each millisecond over a full second's worth of signal and the average correlation peak was measured. This was done while squaring the peak values, in Figure 30, and not squared, Figure 31. The plots show that the noise is reasonably constant around the values 2760 and 46.4 for squared and un-squared results respectively. Following division by the receiver bandwidth, β, these values equate to N_0 and are used in all subsequent calculations that require an estimate of noise level.

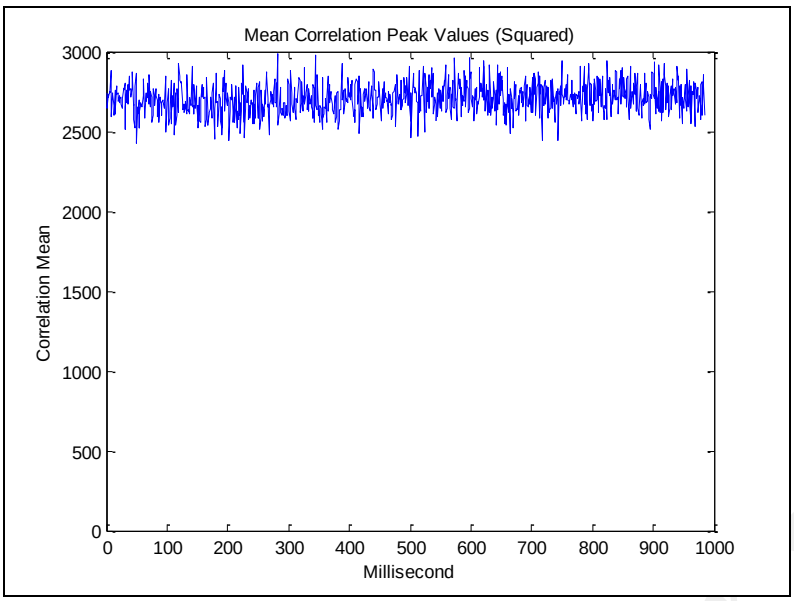


Figure 30. Measured Millisecond Noise Floor (squared)

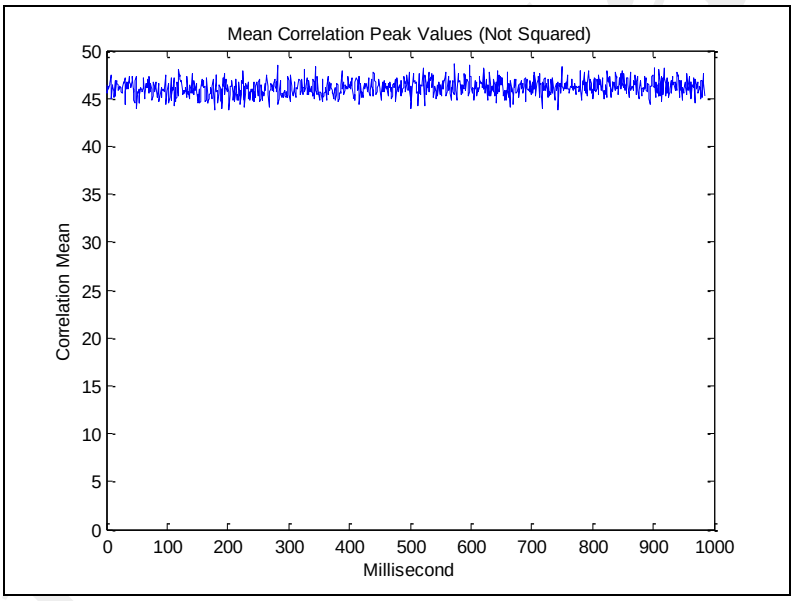


Figure 31. Measured Millisecond Noise Floor (Non-squared)

Using what was now known about the expected noise level, equation (15) could then be used to quantify the strength of the signals detected in terms of C/N_0 (Carrier-to-Noise in dB-Hz).

$$\frac{C}{N_0} = 10 \times \log 10 \left[\frac{P_s}{\beta \times N_0} \right] \quad \textbf{(15)}$$

Where;

β: Bandwidth (Receiver Front-End Bandwidth)

P_s: Signal Power (Correlator Output)

N_0: Noise Power (Derived empirically)

4.1.5 Correlator Accuracy

Around 1200 snapshots were taken using a TrackTag in a fixed position. The signal Dopplers were predicted using an algorithm that would decode the Rinex Ephemeris files found on the internet and by using the snapshot time and location would provide the Doppler and range information for all visible SVs.

Frequency Estimation

All signals experience an equal frequency drift due to the unknown bias related to the tag's RF down-converter's oscillator drift. For this reason, the strongest signal measured by the correlator was assumed to have zero error and was then used to provide the necessary Doppler offset to account for the oscillator drift on the other signals. The difference between the other signals measured frequency against their predicted frequency based on Ephemeris calculations was then regarded as being the frequency error.

Figure 32 shows a histogram of the frequency error.

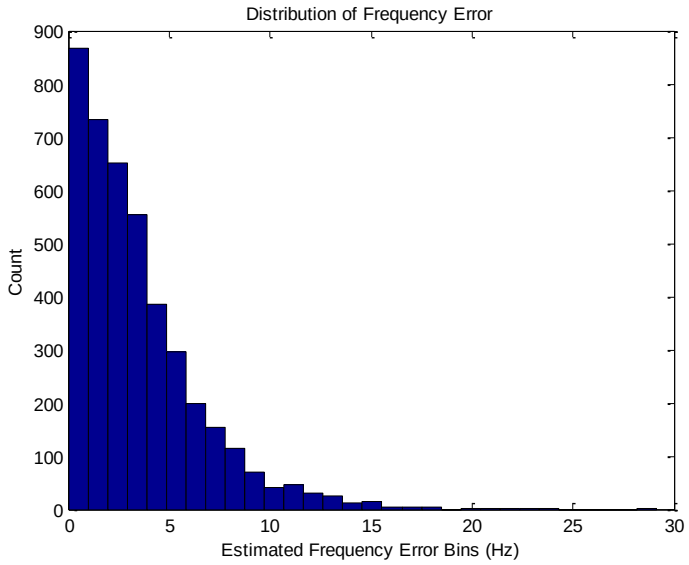


Figure 32. Estimated Frequency Error Using Correlator

Code Phase Estimation

The code phase of the incoming signals is unknown as it depends on both the range to the signal's origin (therefore position) and also the exact point in time relative to the signal's millisecond (or BPSK code) epoch. In order to estimate the accuracy of the detected signals in terms of code-phase, the strongest signal's code-phase was used as the reference and it's offset from its predicted code-phase used as the normalising offset for all other signals. The estimated code-phase error was therefore the difference between the measured code-phase of each of the other signals relative to their predicted code-phase after having the assumed common time offset removed.

Figure 33 shows a histogram of the code-phase error. It is given in metres as the time of transmission can easily be converted to distance using the speed of light.

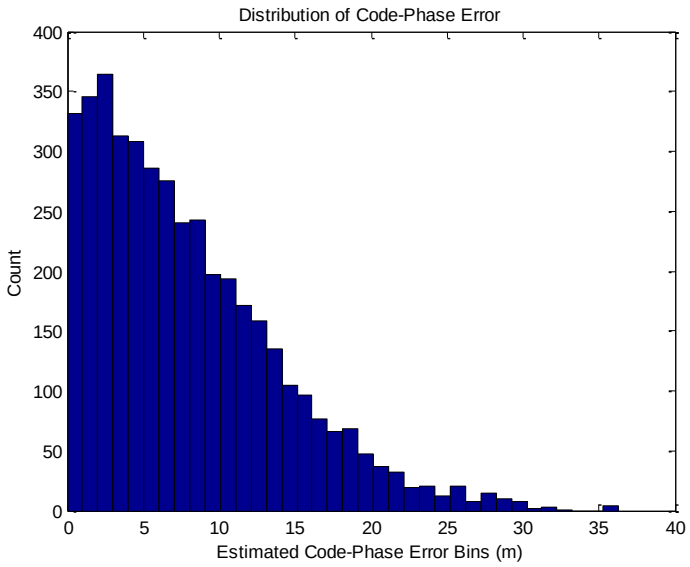


Figure 33. Estimated Code-Phase Error Using Correlator

4.2 FFT Technique

4.2.1 Overview

Although the correlator approach is a very common and well understood method of demodulating BPSK there are major difficulties encountered when trying to use that technique with weak signals. The problem is that a correlator will "lock on" to the first correlation it sees above whatever threshold is set. This makes "false-locking" a major issue when looking at pushing the detection threshold as low as possible. While a correlator searches through each possible code-phase it will stop at the first peak it sees above the threshold and it therefore cannot make a judgement based on the complete picture.

In order to detect weak GPS signals, the topic of the next chapter, convolution of the incoming signal against the expected code for each satellite is required rather than correlation. Using convolution effectively means correlating the signal with all code-phases. So, a vector, equal in size to the length of the signal in samples, of correlation peaks is generated giving a representation of all possible peaks. It is really performing an exhaustive code-phase search. Circular convolution of the GPS gold-codes against the received signal provides a peak at the point where the two signals are in phase. The height of the peak is a measure of the signal power just as with the correlator. The position of the peak is a measure of the code-phase.

Computation time is a major factor when performing the DSP required. For the full circular convolution, every sample (from received signal and its corresponding gold-code point) is multiplied together. This is a complex multiply as the signal has magnitude and phase. The gold-code is then rotated by one point and the same multiply function is carried out. This is repeated until the gold-code has been rotated in full. The number of multiplies required for convolution (M_{conv}) can therefore be derived using equation (16) for the given number of samples (N_s).

$$M_{conv} = N_s^2 \tag{16}$$

With a TrackTag's default sampling rate of 5MHz and the default snapshot length of 26ms, to perform the convolution operation for just one satellite would take 650 million

complex multiplies. That, even by today's standards, is excessive processing. If you also factor in the number of satellites for each snapshot is likely to be around 10, and perhaps 10 Doppler bins are searched, 65 billion complex multiplies would be required for each position fix!

For this reason, most GPS systems involving weak signal detection use an FFT process which utilises the property of time-frequency duality between convolution, in the time domain, and multiplication, in the frequency domain. By taking the Fourier transform of both the signal and the gold-code, multiplying them together and finally taking the inverse Fourier transform, the convolution result is obtained (Figure 34).

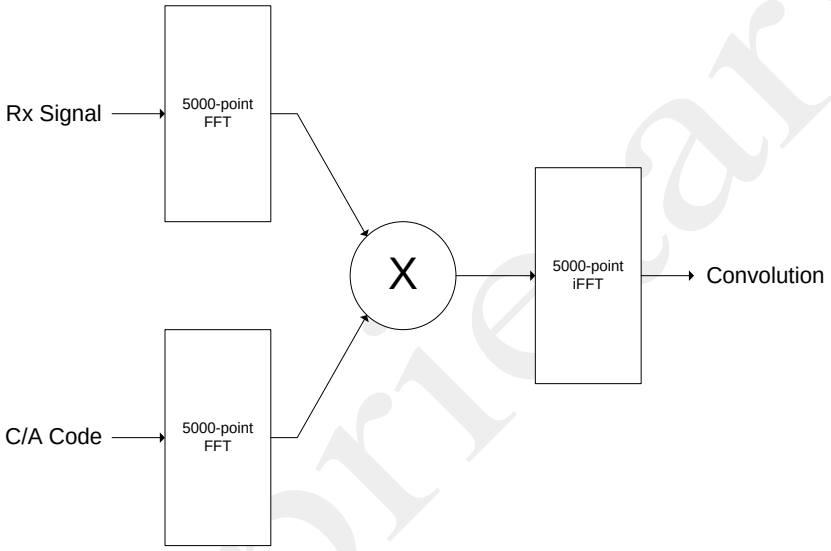


Figure 34. Basic FFT Process

The number of complex multiplies using the FFT technique is derived as;

$$M_{FFT} = N_{samples} \left\lfloor \log_2(N_{samples}) + 1 \right\rfloor \quad \textbf{(17)}$$

For the same snapshot as before this technique requires around 1,890,000 complex multiplies per satellite. This is an extremely small fraction of the processing required and explains why the FFT technique is used. The term "FFT Technique" is used throughout the industry (and in this thesis) to describe the convolution process only because FFTs are used to make the processing more efficient. Strictly speaking, in terms of what the processing is doing it could be described as the "Convolution Technique".

4.2.2 Real-World Signal Detection

The detection function had to be tested against real-world data. The same strong signal used in the correlator analysis in the previous section was used to examine how the FFT process performs. Using the strong signal allowed the signal peak to be seen even over just a single BPSK code period (1ms) and was therefore ideal for developing the algorithm in the initial stages.

Figure 35 shows the FFT output for one millisecond while using an offset of frequency that provided a near zero Doppler offset. A signal can clearly be seen and, when zoomed in, the auto-correlation shape can be seen on the lower plot.

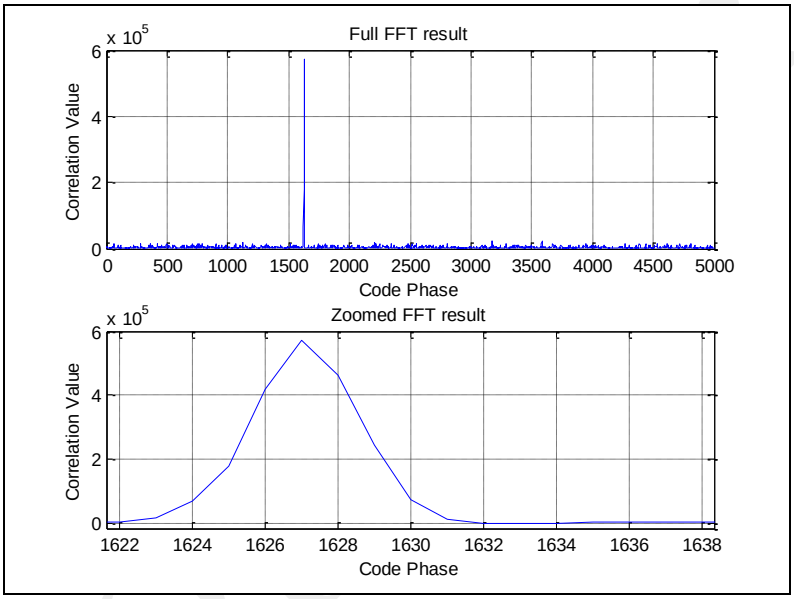


Figure 35. 1 ms FFT Output

By taking the peak value from each one of the Doppler bins processed, the de-spread signal spectrum can be seen in Figure 36. It is not appropriate to plot the signal power surface, as was done with the correlator for Figure 26, as the FFT does not require a code-phase search. This result is plotted as C/N_0 following similar scaling as described by equation (15). The general shape, shows something like the expected $sinc^2$ function [60] with a signal bandwidth of around 1kHz. The bandwidth is related to the integration length. As only 1ms worth of data are taken at a time here, the bandwidth is theoretically equal to the inverse of that integration length, i.e. 1kHz.

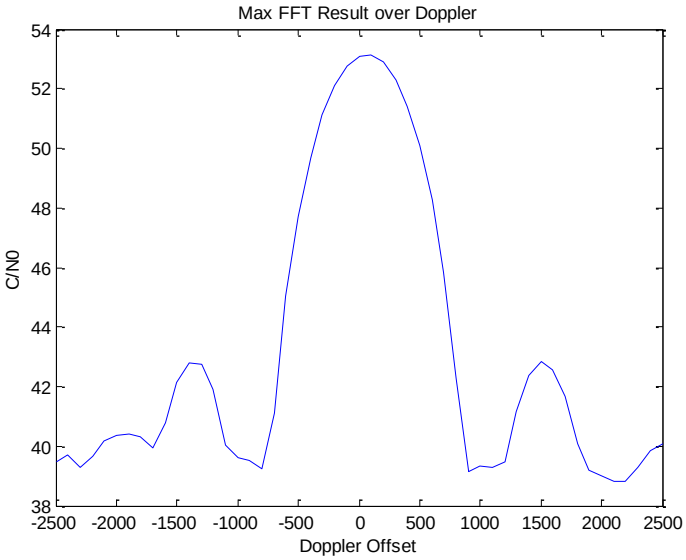


Figure 36. Typical FFT Output over Doppler

Of course, the signal snapshot used was not only 1ms long, typical TrackTag snapshots are around 26ms long (32kB of 2-bit data sampled at 5MHz). The same FFT function is used on multiple ms of the same snapshot to give the results seen in Figure 37. This plot demonstrates that the curves are more-or-less overlapping each other which suggests the processing is as expected. If there were any problems in the algorithms developed it is likely that these curves would not show such consistency.

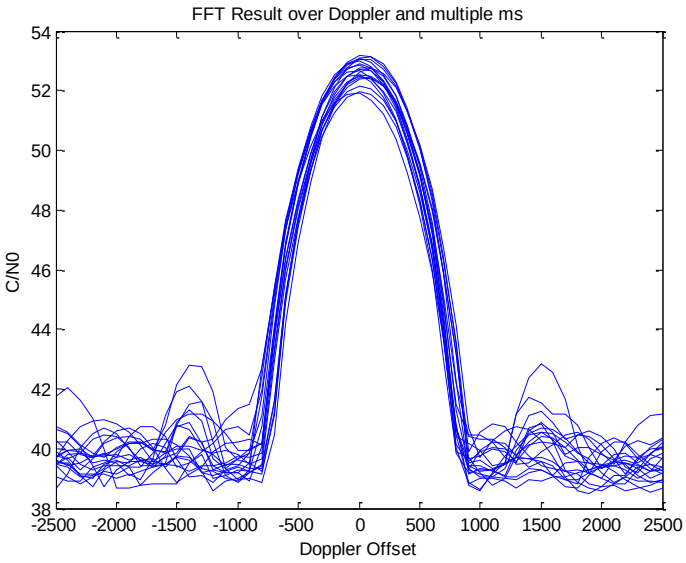


Figure 37. Typical FFT Output over Doppler and milliseconds

4.2.3 FFT Quantization

The code-phase solution provided by the FFT technique is quantized to sample interval length. This is due to the peak being associated with one of 5000 points (N_s) output from the FFT process which operates on the number of samples equating to 1 BPSK code period. So, that is nominally 200ns or 60 metres (when you consider the signal will be travelling at around c = $3x10^8$m/s).

The pseudo-range measurement resolution is therefore around 60m and could certainly have an impact on the accuracy of the measurement. A technique for estimating the sub-sample offset was therefore developed.

The peak resulting from detecting a signal using the FFT is not a discrete peak; there is a predictable pattern which is spread over +/- 1 BPSK chip. This equates to around +/-5 samples over which the autocorrelation function dictates that the signal power is seen even on code-phase offsets neighbouring the code-phase related to the maximum peak value. This was observed in Figure 35 where the signal peak is spread over 5 samples either side of the peak maximum.

The fact that some signal power can be found either side of the maximum peak is made use of in order to estimate the fractional code-phase adjustment required to find the code-phase that is not confined to the discrete sample points (Figure 38).

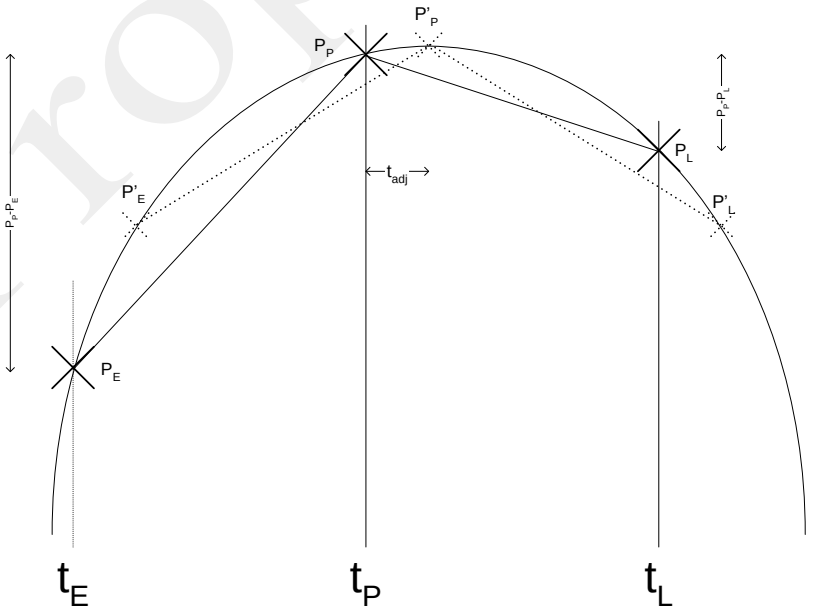


Figure 38. Linear Sub-Sample Adjustment

The actual sample values taken are P_E, P_P and P_L (Early Power, Prompt Power and Late Power respectively) and occur at times separated by the sampling period. The method for estimating the time offset (t_{adj}) required is explained in equation (18) which gives the offset required normalized to +/- 0.5 sample period (of course, the actual peak should be within 0.5 samples of the measured peak).

$$t_{adj} = \frac{2}{\pi} \tan^{-1}\left[\frac{P_P - P_E}{P_P - P_L}\right] - \frac{1}{2} \tag{18}$$

To study the effect of the fractional code-phase adjustment the residual error and the navigation solution (algorithm described in Appendix B) for these points were analyzed. Some data was taken at a fixed location and over a very short time (100 positions over 200 seconds). The navigation fixes therefore should be constant and there should be very little error due to time inaccuracy as the tag clock drift is assumed to be negligible.

The navigation residual error seen, over these 100 snapshots, before the sub-sample adjustment is made was an average of 16.5m, whereas when the adjustment is made the mean is reduced to 6m (Figure 39).

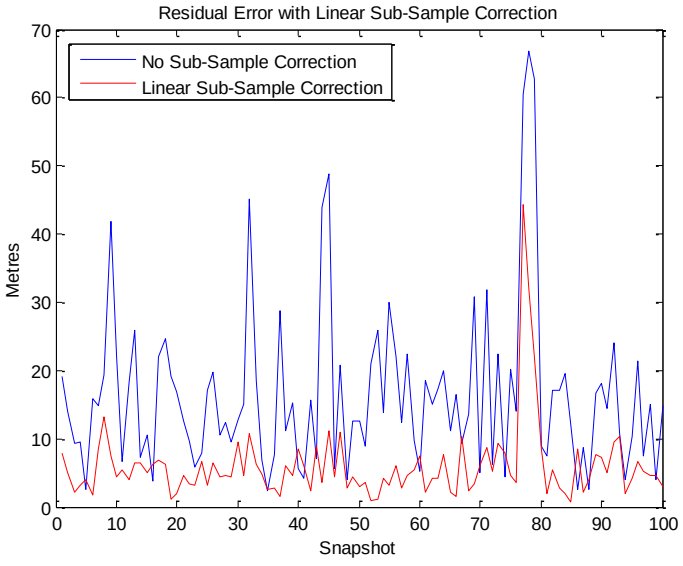


Figure 39. Residual Error with Linear Sub-Sample Correction

The horizontal error is plotted in Figure 40, using the simple navigation solution algorithm, and also shows a noticeable improvement when the sub-sample correction is applied.

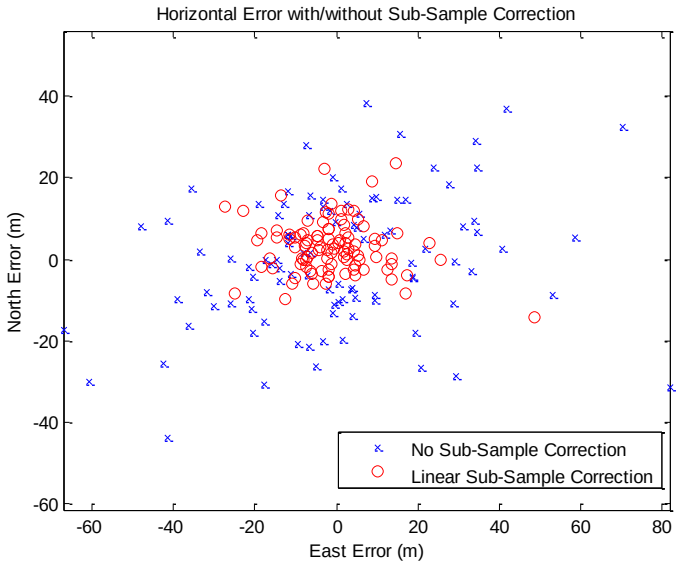


Figure 40. Horizontal Error with/without Linear Sub-Sample Correction

4.2.4 FFT Accuracy

The same 1200 snapshots taken during the correlator tests were used again in tests on the FFT in order to enable comparisons to be made. The signal Dopplers were predicted in the same way, using an algorithm that would decode the Rinex Ephemeris files found on the internet and by using the snapshot time and location that would provide the Doppler and range information for all visible SVs.

Code Phase Estimation

Unlike the correlator, the FFT process did not need to have an estimate of code-phase (within 1 chip of the true value), or a code-phase search, in order to converge on a result. The FFT effectively performs a full code search implicitly. The process, however, does not track the code-phase as the DLL within the correlator does and therefore an amount of degradation in the measurement is expected.

An estimate of the accuracy of the detected signals in terms of code-phase measurement using the FFT was derived using the same method as in section 4.1.4. Figure 41 shows a histogram of the code-phase error when no sub-sample adjustment (as described in section 4.2.3) was made. It is noticeably poorer accuracy than the correlator code-phase error (Figure 33) as the 3dRMS error is around 90m compared to the 30m seen with the correlator.

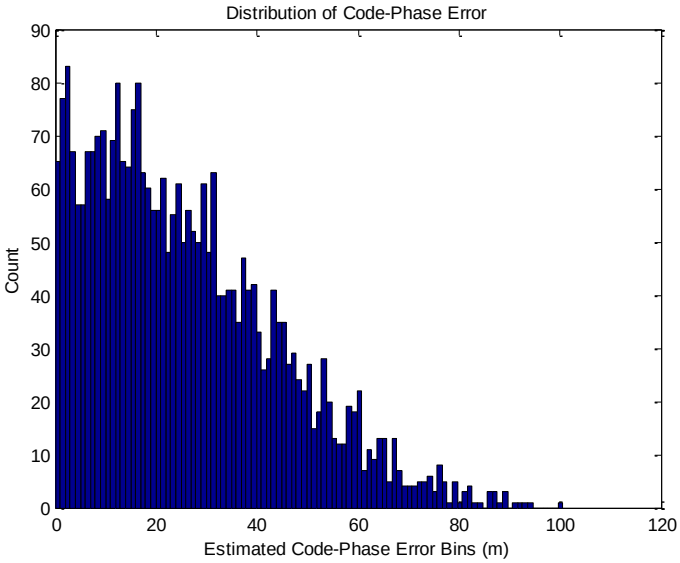


Figure 41. Estimated Code-Phase Error Using FFT

The sub-sample adjustment was then enabled and the result was a clear improvement in the accuracy of the FFT process as seen in Figure 42. The 3dRMS accuracy with the sub-sample correction applied was comparable to the 30m achieved using the correlator

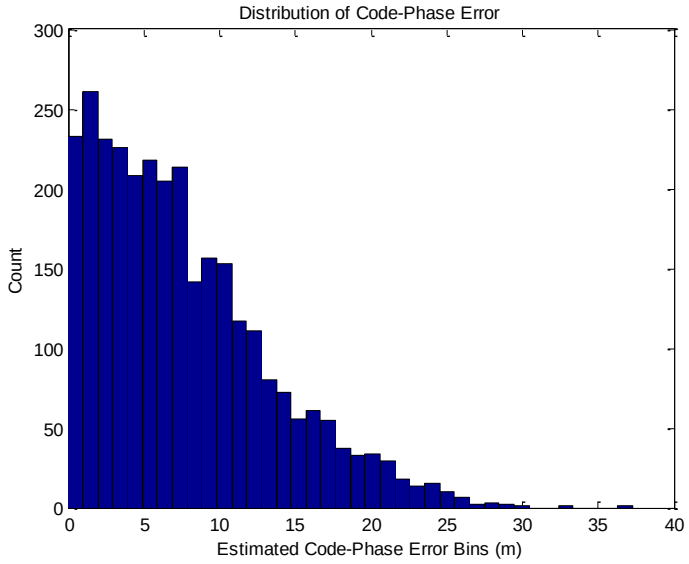


Figure 42. Estimated Code-Phase Error Using FFT with Sub-Sample Adjustment

Frequency Estimation

The FFT process alone does not measure frequency per se; however, an indication of frequency can be derived from the FFT results over a number of Doppler frequency bins. An adjustment, similar to that made to the code-phase, was therefore made to estimate the "sub-Doppler bin" offset and was applied to the FFT algorithm as a whole. The resulting histogram of the frequency error (Figure 43) indicates that the FFT frequency estimation was still significantly worse than that achieved when using the correlator.

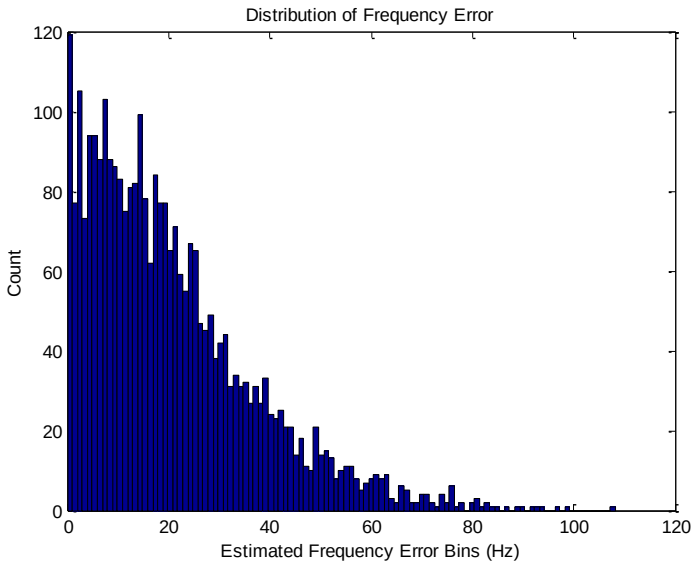


Figure 43. Estimated Frequency Error Using FFT with Sub-Doppler Bin Adjustment

4.3 Comparison of Techniques

Both the Correlator and FFT approaches to GPS signal demodulation have been implemented and assessed. Each has their strengths and weaknesses and these have been demonstrated earlier in this chapter.

The correlator technique consists of tracking loops for code-phase, frequency and carrier-phase. Its key strength is accuracy of measurement. It demonstrated far better frequency measurement than the FFT process. With further development the correlator could be improved to offer more accuracy as the tracking loops were not adjusted in terms of integration lengths. If this were to be done the tracking loop bandwidths could be narrowed and greater accuracy achieved.

The biggest weakness the correlator has is that it is necessary to perform a code-phase search. The correlator will only converge on the signal if the initial estimate of code-phase is within 1 chip of its true value. In fact, in many of the tests analysing the correlator, the FFT process was run prior to the correlator in order to get a suitable code-phase seed to allow the correlator to run.

It is well documented that a correlator-based GPS receiver suffers greatly from false-locking when weak signal detection is attempted. It is the fact that a correlator will stop once it detects a signal above the threshold set that makes it very difficult to detect weak signals without locking on to false peaks. The FFT technique does not suffer this to the same extent due to the full range of possible code-phase being analysed every time a peak detect decision is made.

As the FFT process demonstrated accuracy close to that achieved by the correlator, and with the difficulties faced when attempting to acquire weak signals using a correlator design, it was concluded that the research would focus on development of the FFT-based signal detection.

Chapter 5: Weak Signal Detection

5.1 Introduction

With the FFT-based signal detection algorithm in place as the core building block of the DSP along with the navigation solution code to enable position estimates, actual data from field trials could be used. Development of the algorithms using real world data was preferred to simulation in order to ensure the research focused on the issues faced by biologists using TrackTag.

It was fortunate that a few studies around the world had agreed to invest in TrackTag during this research phase as it enabled close cooperation between the TrackTag signal processing research and its end users. The TrackTag users at this stage included studies looking at the following species;

1. Albatross (South Georgia, Atlantic Ocean)
2. King Penguin (South Georgia, Atlantic Ocean)
3. Seal (South Georgia, Atlantic Ocean)
4. Leatherback Turtle (Zante, Greece)
5. Tapir (Peruvian Amazon)

These studies had had difficulties in using other tracking technologies for various reasons. The marine studies, which basically cover species 2-4, had major problems getting conventional GPS to work due to the wave-wash scenario preventing an uninterrupted signal for long enough. The Albatross studies encountered a barrier in terms of the temporal resolution that could be obtained using GPS as their flights could last for a number of weeks and so, in order to minimise the loading a small battery would have to be used thus limiting the number of positions that could be attempted.

The most severe problems were encountered by the researchers in the Amazon. Not only did their application suffer from terrible satellite signal attenuation due to the rainforest tree canopy, but the creatures would also be away for months at a time. It was, therefore primarily the Amazon data that was used to aid the research into weak signal detection algorithms.

5.2 Weak GPS Signal Detection Review

5.2.1 Signal Integration

The conventional correlation equations (19) are performed for a number of code phase offsets (code search) and also a number of "Doppler bins" (Doppler search).

$$I = \sum_{k=0}^{N-1} y_k c\left(t_k + \hat{\eta}\, t_k - \hat{t}_s\right) \cos\left[\omega_{IF} - \hat{\omega}_D\, t_k\right]$$
$$Q = -\sum_{k=0}^{N-1} y_k c\left(t_k + \hat{\eta}\, t_k - \hat{t}_s\right) \sin\left[\omega_{IF} - \hat{\omega}_D\, t_k\right]$$
$$P\left(\hat{t}_s, \hat{\omega}_D\right) = I + jQ \qquad \textbf{(19)}$$

Where;

N: No. of samples

y_k: Sample input

c(t): C/A PRN code

$\hat{\eta}$: Chipping rate perturbation due to Doppler

$\hat{t}_s$: Code phase estimate

ω_{IF} : Intermediate Frequency used in front-end mixer

$\hat{\omega}_D$: Doppler estimate

The number of Doppler bins depends on how accurate the receiver's estimate of Doppler offset is. For example, the Doppler error would be small if the receiver had detected signals a short time before which resulted in a good position fix and could be trusted to update the estimate local oscillator drift as well as the estimated position. Conversely, if the receiver had not been able to calculate a position fix for a long time, the uncertainty in the local oscillator error and positional error would mean more Doppler bins would be required. In addition, if the receiver experiences some velocity due to the animal's movement, that will translate to apparent Doppler shift. The magnitude of this additional Doppler shift depends on both the direction and magnitude of the animal's velocity as well as the geometry of satellites at that instant.

The number of code phases can vary depending on the algorithm used. If a signal had been tracked a short time previously and the time elapsed since that measurement is accurate and the position has not changed significantly, an estimate of the expected code phase can be made. Some algorithms will perform an expanding code phase search starting at this estimate and moving out in time until a signal is found. Where this

estimate is not practical the algorithm will search across all possible code phases, effectively performing a circular convolution. ("Circular convolution" defined as the process of performing a series of *N* correlations between 2 arrays of length *N*, while rotating one of the arrays by one between each correlation.)

5.2.2 Block Accumulation

A common approach to performing the correlations on the data is to split the data into a series of blocks [61]. Instead of processing the data as a whole, the integration is arranged into several blocks and are correlated with their results are averaged to give an equivalent result. Equation (20) shows how the algorithm can be re-arranged in order to have integration done in ℓ blocks of length N. This assumes that $\hat{t}_s = t_0 + n\Delta t$ and that $c(k + \hat{\eta}(k) - n\Delta t)$ is a sampled version of the signal's PRN code.

$$z_\ell(n, \hat{\omega}_D) = \sum_{k=N\ell}^{N(\ell+1)-1} y_k c_{k-n} \exp(-j(\omega_{IF} - \hat{\omega}_D)\hat{t}_k)$$
$$z_{sum}(n, \hat{\omega}_D) = \sum_{\ell=0}^{L-1} z_\ell(n, \hat{\omega}_D)$$

(20)

Averaging the blocks can be done in a number of ways. A complete, single block over which the correlation is performed would be expressed using equation (20) with both L and N equal to the number of samples in entire data snapshot, which are then averaged to give the correlation power for the complete data. If applied to the full length of a data snapshot, it does of course carry the high risk of data bit transitions cancelling out the correlation power.

At the other extreme, the block length could be made equal in length to the BPSK code period (1ms). This would mean L would equal the number of milliseconds to integrate over. Summing all block correlations is then mathematically equivalent to the full correlation. It does, however, offer both more efficient ways to do the correlations and also opens up opportunities for migrating data bit transitions as discussed next.

5.2.3 Non-Coherent Data Bit Inversion Mitigation

The major challenge with long integration in GPS is that of the signal's data bit transitions. Although not shown in the equation above, the signal is modulating 50Hz

data and therefore the BPSK code could invert at 20ms intervals. An inversion part way through the correlation will attenuate the result. It would, in fact, completely cancel out any correlation power if it were to occur exactly halfway through the correlation due to each side of the inversion giving equal power but with opposite polarities.

One way to handle the data bit inversions is to square the correlations before they are added. This serves to lose the data inversions as each polarity will end up being positive having been squared. Known as "non-coherent" integration due to the phase information being lost through squaring, its use is widespread in GPS. The problem with using non-coherent integration is the "Squaring-Loss" that is encountered. Where coherent integration would offer an improvement of 3dB for every doubling of the integration length, non-coherent integration will offer only 1.5dB [62].

A new technique using a different squaring method in a non-coherent integration has been developed [63]. This appears to offer around 2dB gain overall in the non-coherent processing. However, the technique experiences significant problems when used on signals with even a minor Doppler error. Due to the fact that the TrackTag algorithms are constantly having to estimate Doppler on all signals this technique was discarded for use in the TrackTag research at an early stage. The majority of signal acquisition attempts were assumed to have a large Doppler error, especially were non-coherent processing would be use, i.e. as the first step to acquisition. If the Doppler is known, coherent integration was considered the best way to detect weak signals.

5.2.4 Multiple Data Bit Coherent Integration

The biggest challenge when integrating coherently is that of the data bit transitions. There are twenty 1ms code epochs in each GPS BPSK data bit [64]. This means that any integration that overlaps one of these code epochs will be severely affected. One way to counter this problem is to reverse all the data transitions. This, however, would require knowledge of the data stream as well as very accurate time of transmission in order to synchronize to the correct point in the data. There has been some work on algorithms that require no a priori knowledge of the data bits received to enable long coherent integration, this obviously relies on obtaining the full broadcast data-stream for every GPS satellite over the time of interest. This potential "Assisted" technique was considered out with the scope of this research due to difficulties in obtaining the

broadcast data and the practicalities of storing such large quantities of data. Another issue with assisting using data in this way is the requirement for time to be known with sufficient accuracy to line up the bit edges. The TrackTag system was not capable of meeting such requirements at this point in the research.

Non-assisted GPS coherent integration is often limited to 10ms due to the 20ms code epochs [65]. A period of 10ms is used so that it can be guaranteed that no data transitions will occur in either the odd or even integration blocks. This approach is very commonly used in GPS receivers by taking multiple "Alternate Half-Bits" [66] and is described by equation (21). The 10ms blocks are squared as the polarity between each is unknown. By calculating the "odd" and "even" results of such a process and using the set that offers the higher power significant improvement in SNR can be achieved over pure non-coherent methods.

$$z_{even}(\hat{\omega}_D) = \sum_{a=0}^{N}\left[\sum_{\ell_0=20\times a}^{\ell_0+9} z_\ell(\hat{\omega}_D)\right]^2$$

$$z_{odd}(\hat{\omega}_D) = \sum_{a=0}^{N}\left[\sum_{\ell_0=10\times a}^{\ell_0+9} z_\ell(\hat{\omega}_D)\right]^2 \quad \textbf{(21)}$$

Where;

ℓ : Millisecond Block index

z_ℓ : Millisecond Block correlation

N: Number of millisecond periods to integrate

TrackTag usually only has around 24ms in total available for processing, limiting to integration to only one 10ms block. Use of less than half the (already) very short snapshot of data would present a severe limitation in the weak signal detection process.

There has been research into the use of sliding 20ms correlation windows that attempt to find a GPS bit edge to enable extended coherent integration [67]. The method presented where 1ms blocks integrations have their polarities successively inverted (twenty times to cover each of the 20 possible millisecond phases) to cater for any bit phase alignment, requires no data in addition to the first 20ms. This would therefore fit well

with the requirements of TrackTag although it does not allow for acquisition of the signal using integration over more than 1 GPS bit (20ms).

The concept of the "BACIX" approach [67] was developed during the same period that this research was conducted and has much in common in terms of the general concept. However, the research presented here was done independently and resulted in a subtly different method which has key benefits to the TrackTag system. With TrackTag snapshots being taken on a tag with no processing intelligence for making decisions over snapshot length or frequency, the acquisition algorithm has to make best use of what is offered by the dataset. This means that acquisition should aim to integrate over the entire snapshot where possible.

The "BACIX" approach suffers from the limitation that only the bit phase is adjusted and therefore it cannot handle anything other than one bit transition. If, in the TrackTag system, the user may decide to compromise battery life in favour of taking snapshots many times the length of normal TrackTag snapshots. This would be done if the user believes the signal environment is going to be very tough for the tag to acquire signals. Under that circumstance, integrating over only the first 20ms to acquire would mean the longer snapshots offer no benefit. Integration over multiple GPS bits is a definite requirement to the success of TrackTag's weak signal detection capability. It is therefore considered a primary objective of this research as it was assumed that minimising the signal detection threshold as far as possible would be a pre-requisite to weak signal tracking on animals in harsh environments.

5.2.5 Cross-Correlation

As the noise floor of the signal detection process is pushed down, the effect of "Cross-Correlation" has to be considered. Cross-correlation is the term often used to describe the process whereby the incoming signal is "cross-correlated" against the expected gold-code in order to de-modulate the signal – the basic GPS signal detection process. In this thesis, however, "cross-correlation" is assumed to describe the unwelcome effect of gold-codes correlating well enough against others to cause problems.

Although the BPSK sequences selected for the gold-codes have been chosen to be orthogonal and therefore minimise cross-correlation properties, they are not perfect. The effect is that a strong signal can also show up as a weak signal while looking for a

completely different signal, i.e. using a different gold-code.

There has been some research into the cross-correlation properties of GPS signals [48] which suggest the worst case cross-correlation peaks relative to the source signal strength. The worst case is a cross-correlation peak at -21.6dB down from the strong signal peak.

Many examples of cross-correlation have been found in the TrackTag datasets. An example of cross-correlation interference on a signal taken under the Amazon Rainforest where there was a strong signal present with a Carrier-to-Noise Density of just over 51dB-Hz (PRN-22) is shown in Figure 44. It shows the cross-correlation effect on weak signals in the same snapshot of data. The plot is the result of attempting to track a signal expected from an SV (PRN-1) that was behind the Earth at that time and therefore definitely not visible. While the noise floor was close to the expected value when using non-coherent processing, when pushing the noise floor down further, by employing coherent integration, sporadic peaks are seen around 29dB-Hz well above the expected coherent noise floor.

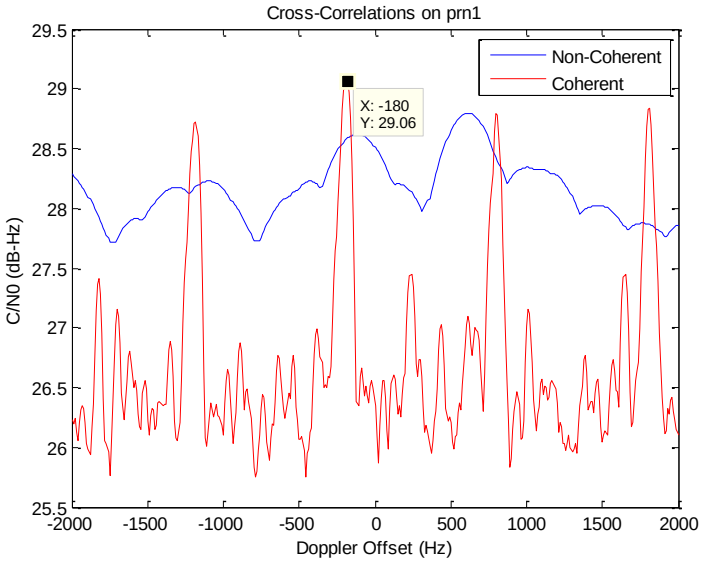


Figure 44. Cross-Correlation from PRN-22 (Satellite provide 51dB-Hz Signal) on PRN-1 channel (Satellite not in view)

The worst case scenario of having cross-correlation -21.6dB below the level of the strong signal means that the example above, with a strong signal of around 51dB-Hz, should exhibit cross-correlation peaks no higher than 29.4dB-Hz. It seems to match what has been observed in practice. This effect should therefore always be considered

while analysing weak signals and care must be taken to ensure there are no strong signals present that would give erroneous measurements for the signal you are actually looking for.

In the TrackTag processing a dynamic signal detection threshold is adjusted depending on the highest signal power detected. Although the main signal detection thresholds are set depending on the integration type and length, this can be over-ruled if there is a strong enough signal detected that would require the detection threshold to be raised to prevent false-locking on to cross-correlation products.

5.3 Non-Coherent Integration

The processing developed for TrackTag performs 1ms convolutions. Each convolution generates the vector result (of length 'n') from performing correlations following the detail provided in equation (20) for every possible sample offset, n. The maximum value of 'n' therefore equals the number of samples there are per millisecond.

$$Z_l(\hat{\omega}_D) = \begin{bmatrix} z_\ell(\cdot, \hat{\omega}_D) \\ z_\ell(\cdot, \hat{\omega}_D) \\ \cdots \\ z_\ell(M, \hat{\omega}_D) \end{bmatrix} \quad \textbf{(22)}$$

Equation (22) shows the relationship between the correlations, z_l, and convolutions, Z_l. for a snapshot of, say, 24ms data using 5MHz sampling would result in 5,000 element vector for each millisecond. Z_l would be calculated for l=1,2..24. Of course, the convolutions were performed utilising FFTs.

When integrating all milliseconds together non-coherently, each convolution element is squared before being summed with corresponding samples from neighbouring millisecond convolutions. Figure 45 illustrates how the millisecond convolutions are accumulated using a non-coherent technique. "Non-coherent" implies that the integration does not keep track of phase between milliseconds. The convolutions are squared; in fact they are multiplied by their complex conjugate of themselves, and therefore lose their phase information. This method is therefore not affected by data bit inversions. The end result is a vector Z_T which represents the integration of all millisecond results in an effort to reduce the noise level.

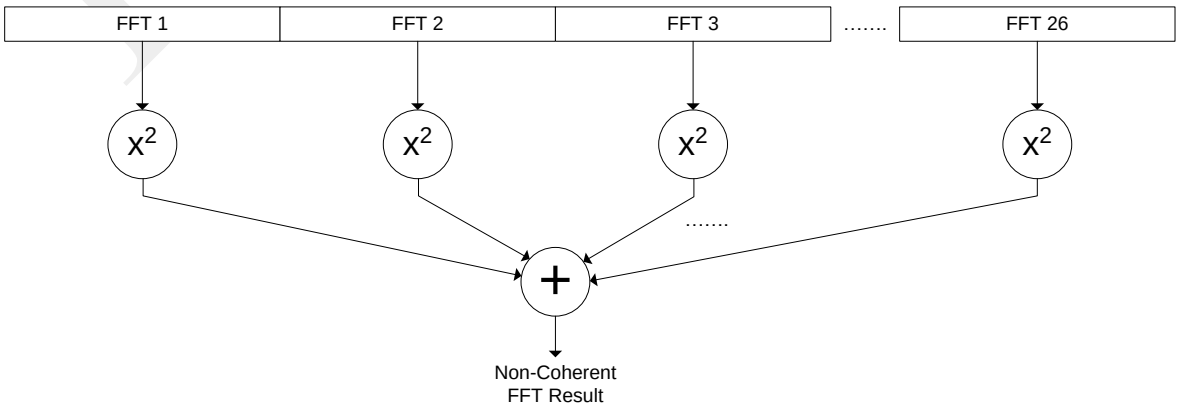


Figure 45. Non-Coherent Integration

The bandwidth of the output signal is related to the length of coherent integration. In the non-coherent case there are multiple 1ms long blocks of coherent integration and therefore the bandwidth of the non-coherent accumulation remains constant at 1kHz.

5.3.1 Carrier-to-Noise (C/N_0) Calculation

The FFT technique takes the raw data and basically performs a convolution of that data against the relevant GPS "gold-code". How this affects the expected noise level is fundamental to the calculation of the noise floor. The output of the non-coherent convolution process, when processing many datasets of noise (shown as blue) and a fixed signal (shown as red), is seen in Figure 46. As the non-coherent integration sums the square of each millisecond, the trend is for the noise floor to rise linearly with every addition. Any signal present would, of course, also rise linearly. The average noise for any integration length can therefore be found by simply dividing the accumulation by the number of milliseconds taken.

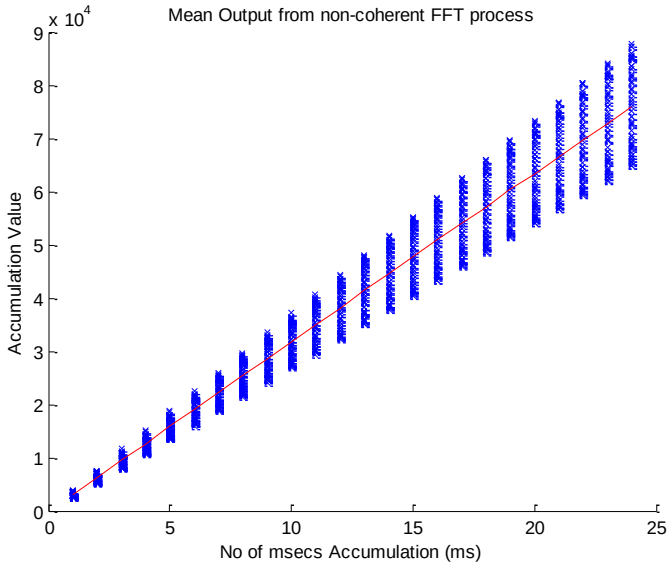


Figure 46. Mean Non-Coherent Convolution Value

In order to compare signal and noise over any signal length, the accumulation value is therefore normalised by dividing by the length of accumulation in milliseconds. The result of this is seen in Figure 47. As expected, the noise level is now constant. Any signal present would also be expected to remain constant following this normalisation. The standard deviation of the accumulation values, however, falls by the rate of the

square-root of accumulation length. It is the reduction in the variance of the noise that enables a threshold to be set closer to the noise floor with less risk of a rogue noise peak giving a false detection.

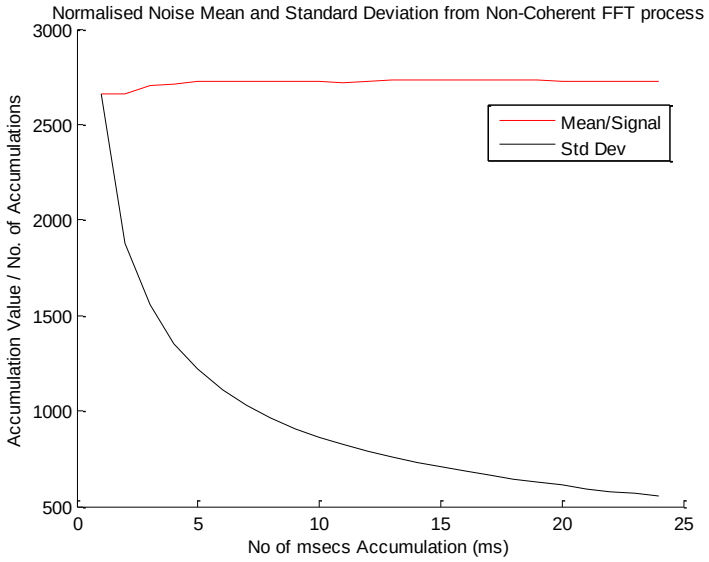


Figure 47. Non-Coherent Normalised Accumulation Mean and Std Deviation

So, the formula used for calculation of C/N_0 for non-coherent accumulation is given in equation (23). The N_0 and β values are constants and P_s is an output from the FFT process along with whatever A_{cc} was used to calculate that result. The expected noise power, N_0, relates to the expected level of noise in 1ms and was determined empirically in section 4.1.4. "$\beta \times N_0$" was actually taken as a constant with the value of 2760 for non-coherent accumulation.

$$\frac{C}{N_0} = 10 \times \log_{10}\left[\frac{P_s}{\beta \times N_0 \times A_{cc}}\right] \quad \textbf{(23)}$$

Where;

β: Bandwidth

P_s: Signal Power

N_0: Noise Power

A_{cc}: Accumulation length (ms)

5.3.2 Thresholding

A large number of snapshots were taken with no visibility of any satellites. This was done to provide a large sample set of noise to enable the following noise analysis to be done.

Figure 48 shows the spread of values, generated from the sample "noise" data, from which 99% of the convolution peaks are below. This equates to a PFA (Probability of False Alarm) of 1%. The spread of values for each accumulation length was seen to be randomly distributed and a rough measure of the maximum was to take the mean and add two times the standard deviation.

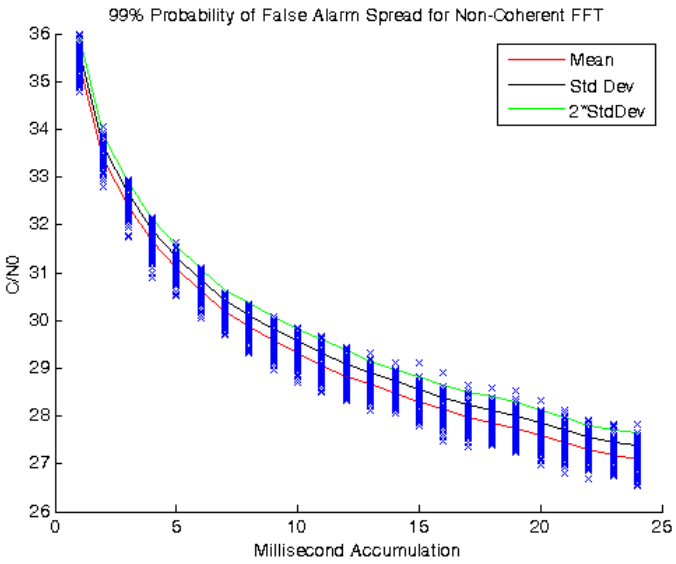


Figure 48. Non-Coherent 99% PFA C/N0

As expected, a drop of 1.5dB per octave in the noise floor with non-coherent integration is observed. This is due to the fact that the noise, although still growing with longer accumulations, has a reduction in its variance that is not growing as fast as the mean. Therefore, the peak-to-mean difference will fall.

Figure 49 shows how the noise floor drops as longer accumulation times are taken for PFA's of 1-5%. The "expected curve" is also superimposed (at an arbitrary level) to show what -1.5dB/octave looks like. The curves follow close to this.

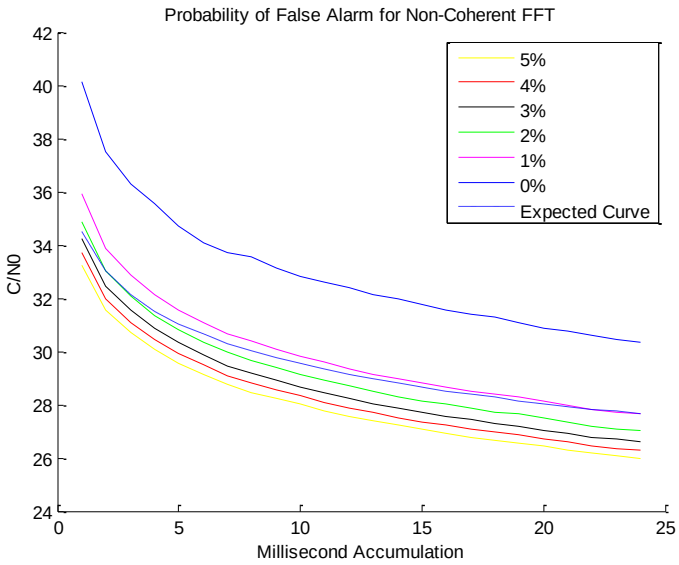


Figure 49. Non-Coherent PFA C/N0

5.4 Coherent Integration

When integrating coherently, each millisecond convolution vector result is simply added to the others (without squaring). Figure 50 illustrates how the millisecond correlations are accumulated using a coherent technique. "Coherent" implies that the phase is preserved between millisecond integrations. This relies on the signal phase being continuous throughout all milliseconds used. This method is therefore affected by any data bit inversions as they invert the signal phase by 180° in BPSK.

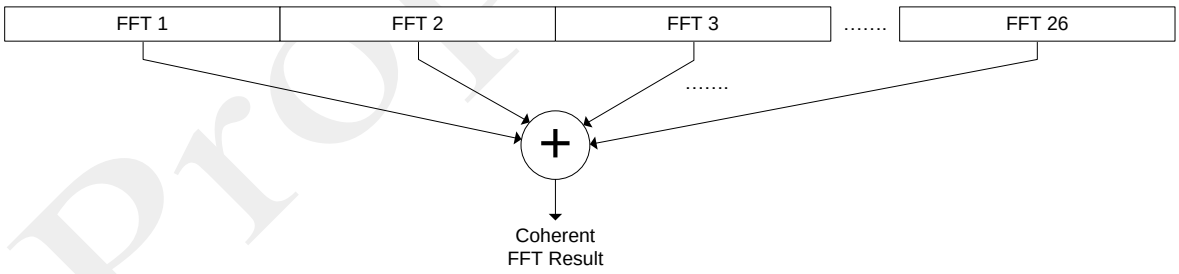


Figure 50. Coherent Integration

As the navigation data superimposed on the BPSK carrier is at 50b/s, data bit inversions can occur at 20ms intervals. Another potential issue is code-phase alignment with coherent integration. If a data bit transition occurs at some point in the middle of the 5000-point convolution, the signal power will be affected. For example, if a data bit edge occurs halfway through the millisecond being correlated, half the signal will be in-phase while the other half will be exactly out of phase therefore cancelling out and leaving zero signal power, even with a very strong signal.

These issues combined meant that coherent processing is always best done when the correct code-phase is used to index into the incoming data in order to force any data bit transitions to either end of the convolution thus minimizing any power loss due to fractional millisecond integration. This makes bit-edge detection (if required) easier due to the expected phase shift encountered at a bit transition being 180° with no intermediate points.

Estimating the code-phase of a signal before it has been processed cannot be done for the 1st signal to be detected in a given snapshot as the exact sub-millisecond offset the signal experiences is unknown (it is of course the very measurement the system is trying to make!). However, once a signal has been detected and the code-phase for that signal is measured, the code-phase for every other potential signal can be estimated based on assumed time and position. This was implemented during the research.

So, assuming code-phase is known, the algorithms would index into the data by the relevant number of samples (which will be different for each SV) and so the correlation peaks were then expected to occur close to the code's rollover point, i.e. sample 1.

The data bits will also be unknown and so the resulting polarity reversal will have a hugely negative effect on the observed signal strength. This is a major topic in many weak signal GPS applications. Designers often use mobile phone technology [68] to aid acquisition of the GPS signals. "Data Aiding" whereby the actual GPS data bits are received from another source, can be used if time is known accurately enough; the data bit transitions can be applied to the incoming signal to aid coherent correlation. This option was not available to TrackTag at the time of this research.

Another option that is found in some text [59] on GPS is to integrate over two successive 10ms windows. This means that you are guaranteed to have one 10ms block free of a bit transition. This was actually implemented for some time during this research work and worked fairly well. The problem with it is that it limits the integration to 10ms. 24ms snapshots have been used almost exclusively thus far but it is anticipated that longer snapshots will be used in the future to provide additional sensitivity in weak signal environments. The two 10ms integration window technique

would not be able to make use of longer snapshots and, of course, there was also the potential to more than double the integration length even when limited to 24ms.

Another issue with coherent integration is the fact that the length of coherent integration affects the processing bandwidth. With the non-coherent processing discussed earlier, the bandwidth was effectively fixed and had nulls at offsets of ±1kHz due to the integration length always being 1ms. With coherent processing these nulls, and consequently bandwidth, are pulled in rapidly as the coherent integration length increases. The result of this is that the estimate for Doppler offset has to be more accurate and/or more Doppler bins have to be taken to ensure the signal is not missed.

A new technique has been developed here for the TrackTag algorithm design. The "All-Bit-Permutation" accumulation technique basically calculates what the signal power would be given all possible bit patterns and selects the pattern with the highest correlation peak.

5.4.1 All-Bit-Permutation Accumulation Technique

The technique developed here is a novel approach to obtaining coherent integration without knowledge of the data bits. It was necessary to design this new algorithm in order to maximise the integration length over the relatively short snapshot lengths TrackTag takes. With such a short snapshot and no chance of making the decision to re-attempt the processing, it was crucial to aim for full usage of the entire snapshot.

Working out all permutations of bit patterns over a certain time interval requires a systematic approach. Not only are the data bit polarities not known, but the data bit phase (in ms) is unknown. This means that there has to be an intermediate accumulation stage as both bit polarity and phase are to be considered.

There are 3 levels of uncertainty of time;

1. BPSK Code-Phase. This is what is detected through measurement, i.e. the signal time-of-reception modulo 1ms. This is also what is used in the navigation solution equations and works provided the assumed position is not in error by >300km.
2. Navigation Data Epoch. If the BPSK code-phase is known, the location of the navigation data epoch will be at an integer number of milliseconds away from

that and repeats every 20ms (whether or not it actually results in a data bit inversion).

3. Navigation Data Pattern. If the navigation data was known (as with data-aided systems) the data pattern is predictable and therefore the bit inversions could be accounted for during the integration.

The navigation data epochs can occur at any one of 20 phases due to the navigation data bit rate being 50Hz and hence 20ms period. With TrackTag snapshots being so short (typically 24ms), most of the bit's recorded will actually be partial bits. Figure 51 illustrates how up to 3 bits (whole or partial) can be present in just 24ms worth of usable snapshot.

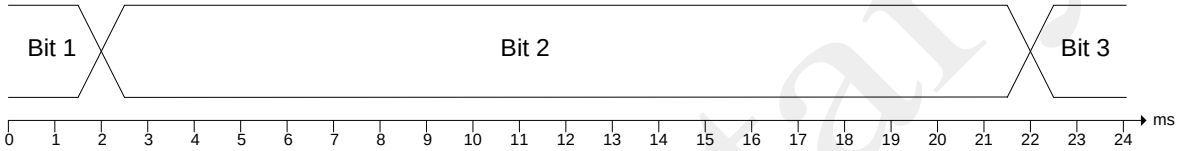


Figure 51. Possible Bit-Snapshot Alignment

If a data bit transition occurs exactly halfway through the 24ms snapshot, no complete data bit will be present. Just as for correlation of 1ms with a data bit inversion halfway through, an inversion halfway through a snapshot would give a signal power of zero even for a strong signal as both halves cancel out.

The algorithm therefore has to calculate results for any possible bit phase. This means the first bit transition can occur anywhere between the 1st and 21st millisecond. If there are 2 transitions, the 2nd would occur anywhere between the 22nd and 24th milliseconds. There would be 20 possible bit phases, every one of which would have to be attempted.

All of this assumes that the data has been indexed into in such a way that code-phase is near zero. With data bit epochs always occurring at zero code-phase, there should never be a bit transition within one (1ms) convolution. This indexing means that we lose a little snapshot length as the data is basically truncated at the start to line up with millisecond boundaries. In order to provide 24ms of usable, complete milliseconds of data, 26ms or data is actually recorded. This requirement is due to the fact that the 1st millisecond worth of data being indexed into and the 26th millisecond is effectively truncated as a result of the same indexing into the data.

An intermediate accumulation matrix is constructed of size 20×(No. of possible bits)×5000 (5000 samples per millisecond as the data is sampled at 5MHz). The number of possible bits depends on the snapshot length, i.e. a 24ms snapshot could have 3 bits as seen previously. The other values refer to the number of bit phases possible (20) and the correlation length in samples (5000).

The objective for the coherent integration is to provide the 5000 element vector that will subsequently be used in the peak detection stage to determine whether (and where) a signal occurs. The intermediate accumulation matrix therefore holds a set of 5000 element vectors. With a 24ms snapshot, there are potentially 3 (full or partial) bits present. There are 20 potential 5000 element vectors the 1st bit could take, depending on the length of that 1st bit as it could be anything between 1 and 20ms long. The 2nd bit vectors are calculated as being a accumulated values assuming that 2nd bit is positioned between 2-21ms, 3-22….or 21-24ms. Of course it is limited to the absolute length of the snapshot. The 3rd bit's potential accumulator values are also calculated. This partial accumulation is described both by Figure 52 and Equation (24).

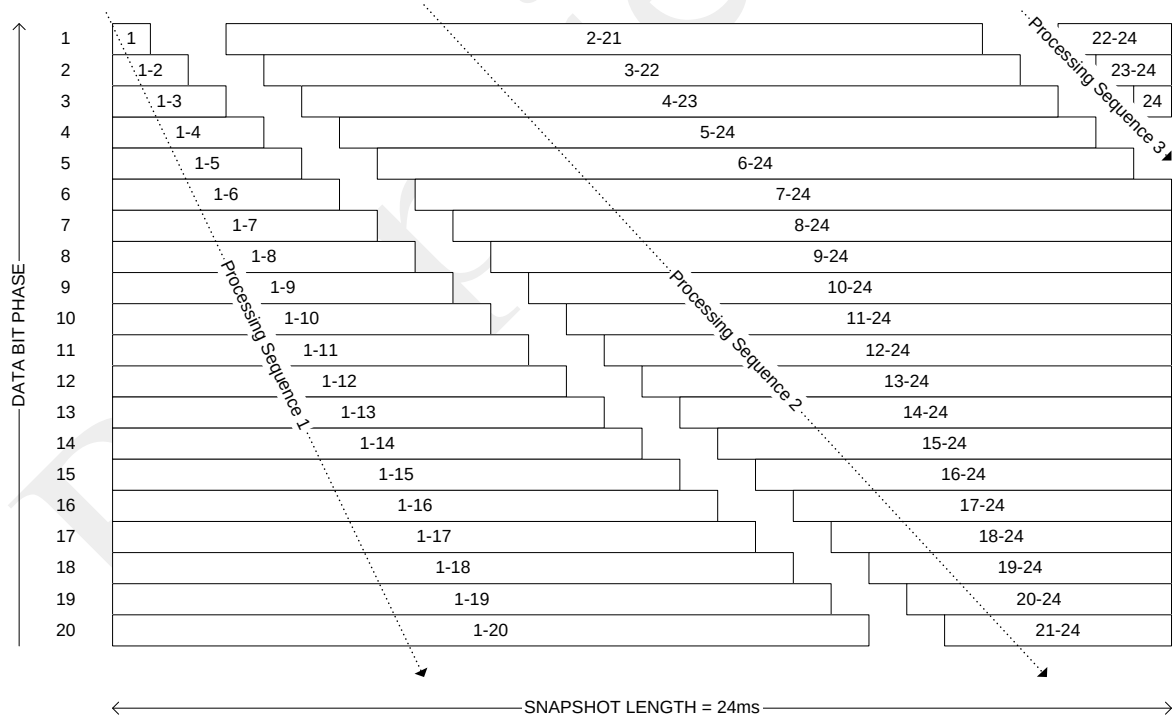


Figure 52. Partial Accumulations

So, for the 24ms snapshot, the accumulation process has to sum 24, 5000 sample vectors for 20 possible phases for each of the 8 possible bit patterns that 3 bits can have. This turns out to be extremely computationally intensive (23×5000×20×8 = 18.4 million

complex additions) and therefore an effort had to be made to minimize the burden placed on the processor.

The intermediate matrix is generated in order to lessen the computational burden. It aims to minimize redundant calculation is described in Figure 52. The fact that accumulations are calculated sequentially by only adding to or subtracting from the previous result means there are no long summations to be performed and every vector addition and/or subtraction stage results in a required accumulator result. The accumulations for each bit (or partial bit) are calculated for all possible bit phases. This is done by adding only 1 millisecond (5000 complex numbers) at a time while storing the resulting accumulation for each data bit phase as shown as "Processing Sequence 1". The next sequence runs through adding the next millisecond until it hits the end of the snapshot. However, it also starts to subtract milliseconds from the lower end when the effective accumulation length is equal to the GPS data bit period (20ms).

So, the outcome of performing this partial accumulation is a 20×3×5000 complex number matrix. It is effectively provides the convolution results for all 20 possible bit phases.

If the convolution output vector for each 1ms is z_l (where l is the millisecond index), the next step is to multiply the partial accumulation matrix by a matrix with half of all possible bit patterns [half of the possible bit patterns are used as the other half is simply the inverse polarity and will provide the same result only inverted].

Equation (24) shows the mathematical description of how the partial accumulations are used, together with the possible bit patterns, in order to generate a matrix containing vectors representing every permutation of accumulation possible.

$$FullAccumulationMatrix = \begin{bmatrix} z_1 & \sum_{\ell=2}^{21} z_\ell & \sum_{\ell=22}^{24} z_\ell \\ \sum_{\ell=1}^{2} z_\ell & \sum_{\ell=3}^{22} z_\ell & \sum_{\ell=23}^{24} z_\ell \\ \sum_{\ell=1}^{3} z_\ell & \sum_{\ell=4}^{23} z_\ell & z_{24} \\ \sum_{\ell=1}^{4} z_\ell & \sum_{\ell=5}^{24} z_\ell & 0 \\ \vdots & \vdots & \vdots \\ \sum_{\ell=1}^{19} z_\ell & \sum_{\ell=20}^{24} z_\ell & 0 \\ \sum_{\ell=1}^{20} z_\ell & \sum_{\ell=21}^{24} z_\ell & 0 \end{bmatrix} \times \begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \quad \textbf{(24)}$$

For the 24ms case, the resultant matrix is 20×4×5000. This relates to 20 possible data bit phases, 4 possible bit patterns (2^3 / 2) and the 5000 samples. The highest value found in the entire matrix will indicate what the bit pattern and phase is as well as the code-phase of the signal. This is an intensive number-crunching exercise and it should be noted that it must be performed for all Doppler bins. Methods for minimising the uncertainties in the front-end oscillator drift and Satellite Doppler estimate are required in order to reduce the overall processing time.

5.4.2 Coherent Carrier-to-Noise (C/N_0) Calculation

The noise floor with coherent accumulation was expected to differ from that seen when using non-coherent integration. This is shown in Figure 53. The coherent integration sums the millisecond result without squaring. Any signal present will grow linearly with accumulation time as with the non-coherent case. The noise, however, tends to self-cancel. The trend for noise is observed to rise only with the square-root of accumulation length.

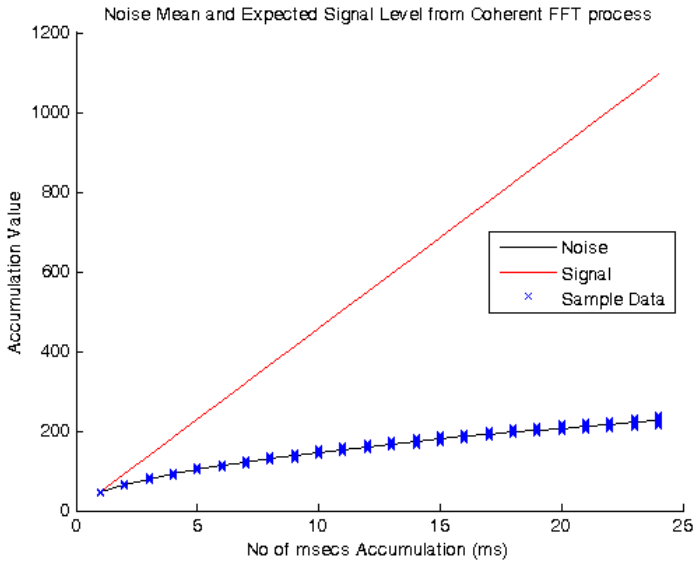


Figure 53. Coherent Accumulation Mean and Expected Signal Level

To normalise the noise, the accumulations are therefore divided by the square-root of accumulation length. The result of this is shown in Figure 54.

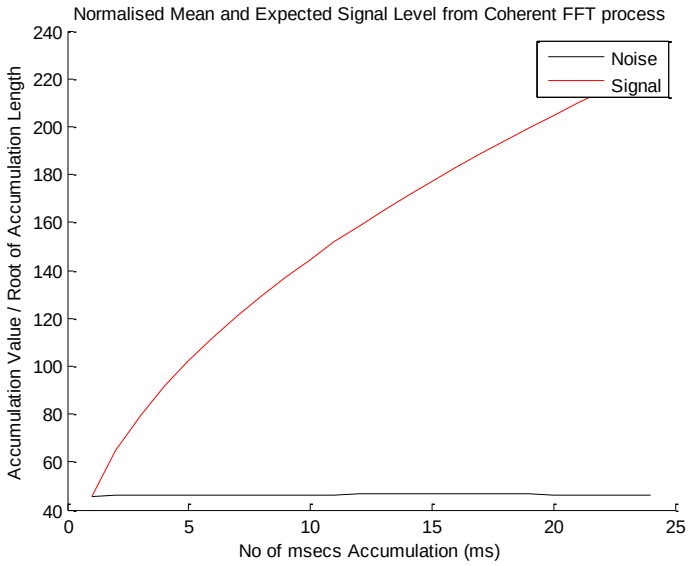


Figure 54. Coherent Normalised Accumulation Mean and Expected Signal Level

Noise is now constant, as expected, but the signal level is still rising. This is due to the signal originally rising linearly and only being divided by the square-root of accumulation length. As C/N_0 should not rise as the accumulation time is extended, there is an additional term required here to correct for this.

$$\frac{C}{N_0} = 10 \times \log_{10}\left[\frac{P_s}{\beta \times N_0 \times \sqrt{A_{cc}}}\right] \quad (25)$$

Where;

β: Bandwidth

P_s: Signal Power

N_0: Noise Power

A_{cc}: Accumulation length (ms)

The expected noise level ("$\beta \times N_0$") in 1ms was deduced empirically (in section 4.1.4) and turned out to 46.3 (very much lower than the non-coherent case due to the accumulation values not being squared)

5.4.3 Threshold Setting

Figure 55 shows the spread of values, generated from the sample "noise" data, from which 99% of the convolution peaks are below. This equates to a PFA (Probability of False Alarm) of 1%. The spread of values for each accumulation length was seen to be randomly distributed and a rough measure of the maximum was to take the mean and add two times the standard deviation.

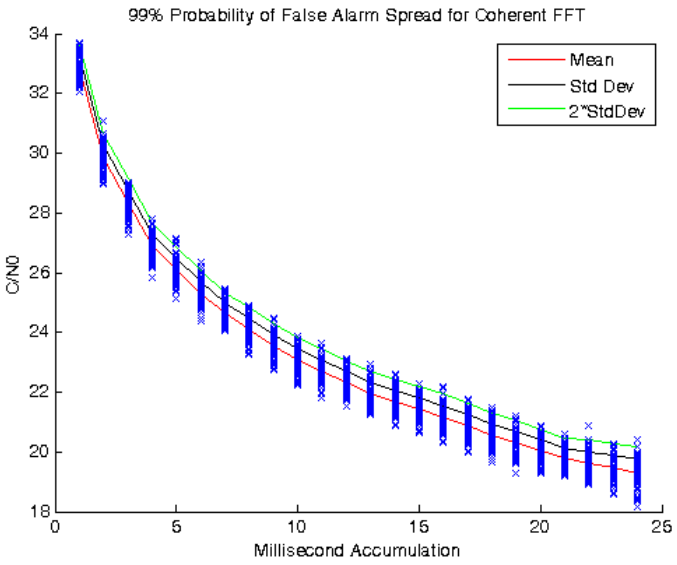


Figure 55. Coherent 99% PFA C/N0

As expected, a drop of 3dB per octave in the noise floor with coherent integration was observed. This improvement over the non-coherent process is due to the fact that the noise does not grow as fast as a signal, it self-cancels.

Figure 56 shows how the noise floor drops as longer accumulation times are taken for PFA's of 1-5%. The "expected curve" is superimposed (at an arbitrary level) to show what -3dB/octave looks like. The curves follow close to this.

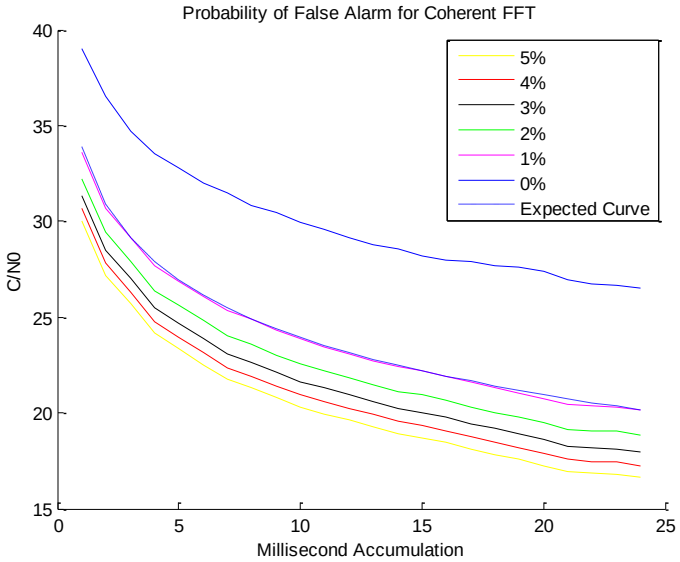


Figure 56. Coherent PFA C/N0

5.5 Peak Detection

5.5.1 C/N_0 Thresholding

This has been discussed to an extent in the sections on calculation of C/N_0 for both coherent and non-coherent integration. In setting a simple threshold C/N_0, above which a signal peak is considered a true signal, it is the PFA (Probability of False Alarm) that is actually of interest. Figure 49 and Figure 56 show how the PFA varies with C/N_0 and integration length when using non-coherent and coherent detection respectively.

Due to the basic navigation solution code implemented for this research, any false measurement will affect the accuracy of the position fix (more advanced navigation solution code would help combat the effects of false signal acquisition but this research does not intend to cover that side of the GPS system). For this reason, the PFA is minimised as far as practical. For example, while using non-coherent integration, the threshold set if integrating over 10ms is 34dB-Hz which is on the 0% PFA curve (Figure 49).

The thresholds are therefore set as constants normalised to an integration length of 10ms. Equations (23) and (25) are then used to calculate what the corresponding threshold should be for the integration length used. This also offers the flexibility to adjust the integration length used without having to also adjust the thresholds.

5.5.2 Peak Noise

Although the C/N_0 thresholding should minimise any potential false lock measurements, an extra check is done which calculates the difference between what is deemed a signal and the next highest peak.

As the signal peak actually covers a number of correlation points based on the Auto-Correlation Function, the samples within 1 chip of the peak are masked out and the remaining highest value is then considered the "Peak Noise".

The extra check on the signal is then defined as a Peak Noise threshold which is the difference between the assumed signal peak and the Peak Noise. This has been set to an arbitrary 3dB during the research which implies that any signal used in the navigation code will be at least twice the height of the next peak in the correlation result.

5.5.3 Cross-correlation Impact on Thresholding

The example previously shown in Figure 44 demonstrated the effect that a strong signal of over 50dB-Hz can have on other signals. When pushing the noise floor down further, by employing coherent integration, sporadic peaks were seen around 29dB-Hz, i.e. well above what the coherent threshold could be set to.

A more controlled experiment was then conducted where a TrackTag was used to take data outside, in good conditions, and then data was taken inside a metal box. The measured noise floor was then compared between data with and data without a strong signal. This was done by trying to correlate against a gold-code that is definitely not present (PRN-12) as that satellite did not actually exist in the GPS constellation at the time.

A separate check on the data found that there was a 51dB-Hz signal present while the device was outside. Figure 57, however, shows how the noise floor jumps significantly depending on whether the tag was inside or outside. This is the correlation power of a signal that is not present and should therefore be equivalent to noise floor. The test taken inside a metal box with no signals present shows a noise level around the expected noise floor, whereas the test taken outside moments later shows evidence of some cross-correlation between codes.

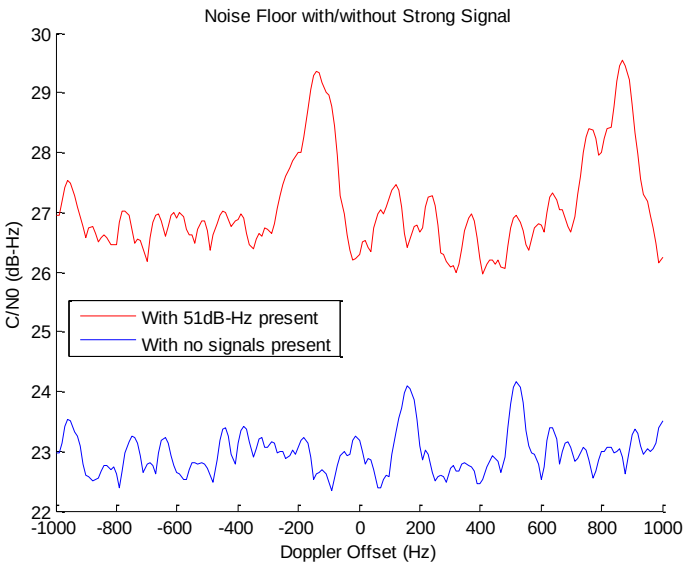


Figure 57. Noise Floor with/without Strong Signal Present (PRN-12 did not exist at that time)

The cross-correlation noise floor is around 27dB-Hz rather than the expected 23dB-Hz. It also seems to vary over Doppler frequency as there are some peaks observed at up to around 29.5dB-Hz.

The worst case scenario of having cross-correlation -21.6dB below the level of the strong signal means that using the previous example with a strong signal of around 51dB-Hz, we should expect cross-correlation peaks no higher than 29.4dB-Hz. It seems to match what has been observed in practice.

In terms of how this affects the implementation of signal detection thresholding, a third check is introduced that uses the highest peak strength and sets a C/N_0 threshold based on that. This relative threshold was set to -19dB as a post-processing algorithm configuration keyword*†*. This value set a relative power level threshold referenced to the maximum signal power and was set to provide some margin (2.6dB) to accommodate some margin for measurement uncertainty above the theoretical -21.6dB cross-correlation worst-case.

†:- Constants used in the algorithms were often defined in a configuration text file that would be read by the code on initialisation in order to allow easy adjustment without having to change values hard-coded throughout the algorithm files. These parameters are referred to as "keywords" in the algorithms.

5.6 Coherent & Non-Coherent Integration Summary

5.6.1 Noise Floor

With non-coherent integration, both signal & noise values are squared as they are accumulated (therefore proportional to power). They both rise linearly with accumulation length. The accumulation value is therefore normalised by dividing by the number of accumulations. This normalized value can then be compared against the expected noise level for 1 accumulation (1ms). The expected noise level has been measured using sample data. This results in a 1.5dB reduction in noise for every doubling of accumulation length and is achieved because the variance of the noise only rises with the root of accumulation length - not linearly.

With coherent integration, the accumulation is done without squaring. So, the signal will still be expected to rise linearly, however, the noise will self-cancel. The noise, in fact rises with the square-root of accumulation length and, after normalizing for that, there is an additional term required in the C/N_0 calculation in order to prevent the signal level rising as it is actually the noise that should be falling. This results in a 3dB reduction in noise for every doubling of accumulation length and is achieved because the noise is falling with respect to any signal.

In order to track weak signals the noise floor was measured using the current non-coherent integration. This was done by processing some 24ms data taken when there were no satellites in view and was found to be around 28dB-Hz (Figure 58). A coherent method was then implemented and the resulting noise floor is plotted on the same graph. A drop of around 10dB is expected due to the theoretical drop of 3dB every time the accumulation period doubles.

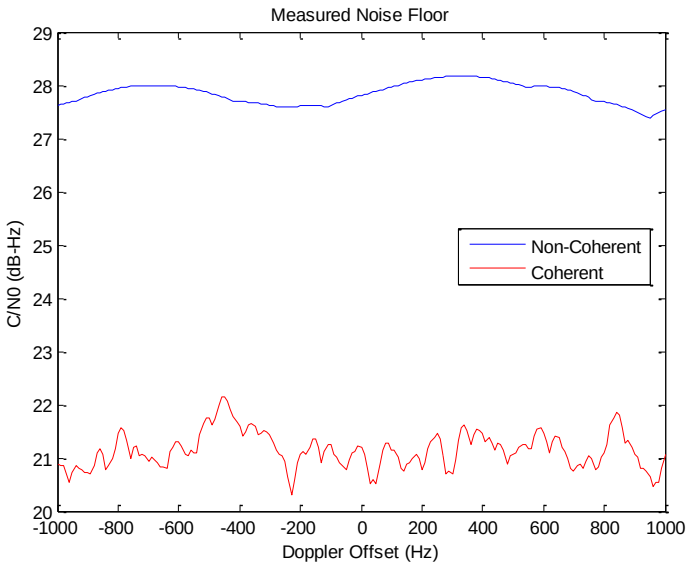


Figure 58. Comparison between Coherent & Non-Coherent Noise Floor

In the case of coherent integration, a 3dB/octave reduction in noise floor is expected compared to the 1.5dB/octave seen while using non-coherent integration. This means there should be around 7dB difference between the noise-floor for each technique over the 24ms snapshot. The drop in noise therefore seems to tie in well with the theoretical performance.

5.6.2 Bandwidth

Figure 59 shows the expected sinc^2 function [48] for both methods using a simulated waveform with no noise. The non-coherent and coherent (over 10ms in this simulation) methods show their null locations at 100Hz and 1kHz respectively. This is also in line with theory.

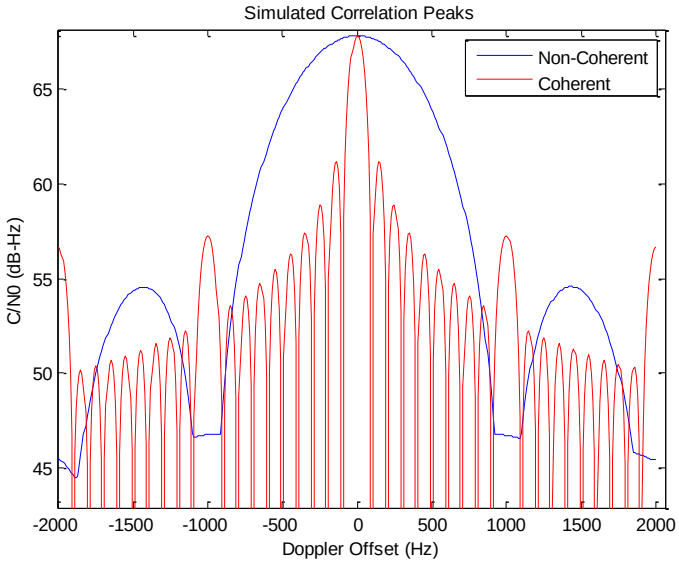


Figure 59. Comparison between Coherent & Non-Coherent Bandwidth

Although simulation of the signal is sometimes useful, real data is the true test. The C/N_0 against Doppler frequency was plotted for SV22 on snapshot 219 of "Trial06102004C" data taken from an Amazon Rainforest trial. This signal data was used as it was known to be strong. Figure 60 shows that the real signal also follows the pattern expected. The bandwidth indicates how much closer the Doppler estimate has to be before coherent integration can be used.

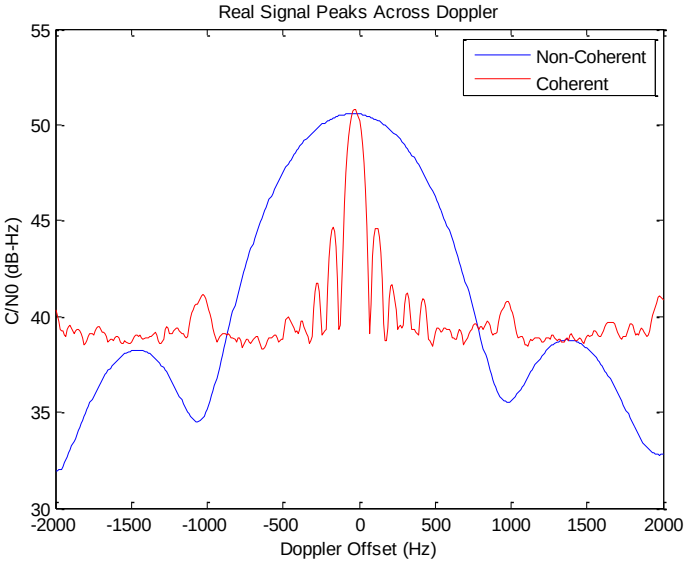


Figure 60. Coherent & Non-Coherent Bandwidth using real signal

This highlights the need for accurate Doppler frequency estimation before an extended snapshot integration is performed as, at longer integration lengths, the nulls of the signal to be detected will occur much closer to the centre frequency of the signal peak.

5.7 Static Point Tests in Amazon Rainforest

The following plots demonstrate why the effort was made to lower the noise floor (and subsequently the detection threshold). It shows how the performance of the navigation solution is improved when using the "All Bit Permutation" coherent integration compared to a non-coherent integration.

Figure 61 shows the advantage the coherent integration offered in terms of capturing signals down to around 29dB-Hz rather than 34dB-Hz at 18 different locations.

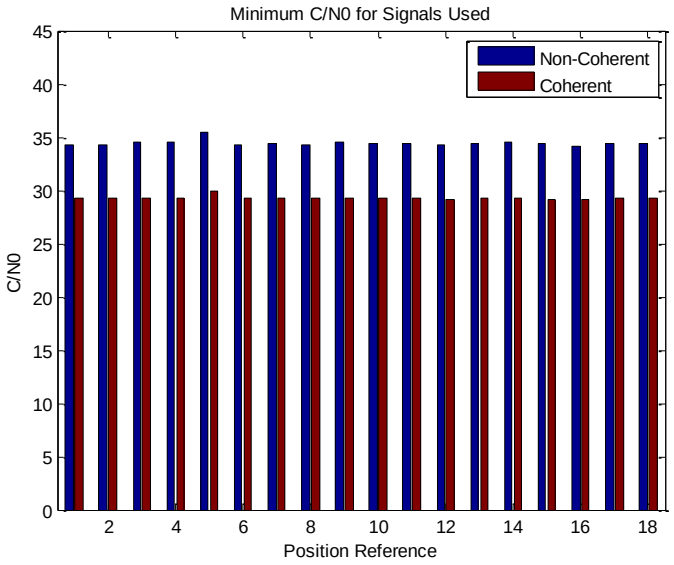


Figure 61. Amazon Static Position Points - Minimum C/N0 Signals Detected

Figure 62 shows the impact of having a lower detection threshold to be an increase in the number of signals detected. This is as a result of the lower minimum C/N_0 achievable with the coherent processing. The increase in observable signals in turn results in a much higher success rate in computing positions as more data is available (Figure 63).

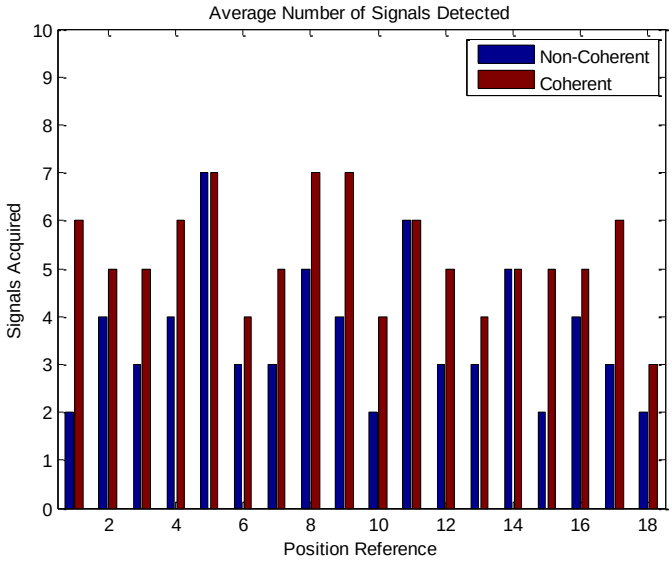


Figure 62. Amazon Static Position Points - Average No. of Signals Detected

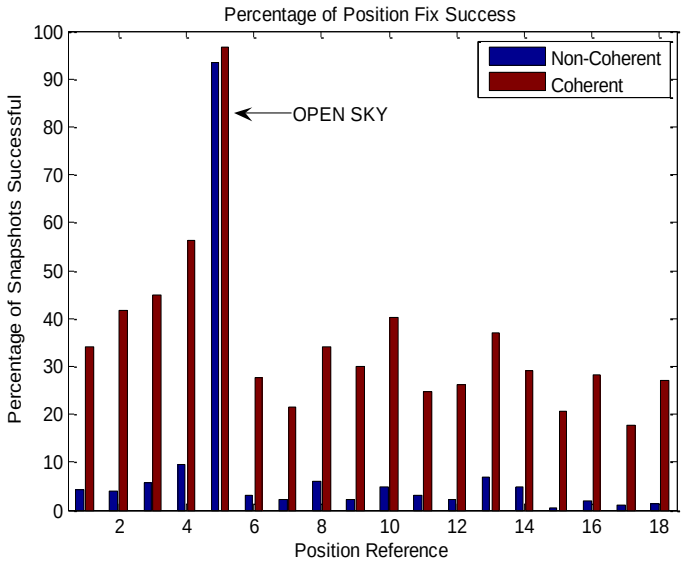


Figure 63. Amazon Static Position Points - Fix Success Percentage

The Reference points are locations that correspond to points measured using a Differential GPS to determine the accuracy of TrackTag. A DGPS receiver was used to estimate the locations to within around 3 metres which is assumed to be negligible in this analysis and so the 2dRMS error calculation for TrackTag fixes (Figure 64) assumes that the DGPS is truth. The locations chosen spanned a range of tree coverage from 65% to 95%. Tree coverage was taken using map data available to the researchers carrying out the biological study. Reference point 5 was "OPEN" i.e. no tree cover which explains why there was a very high fix success rate for both integration methods.

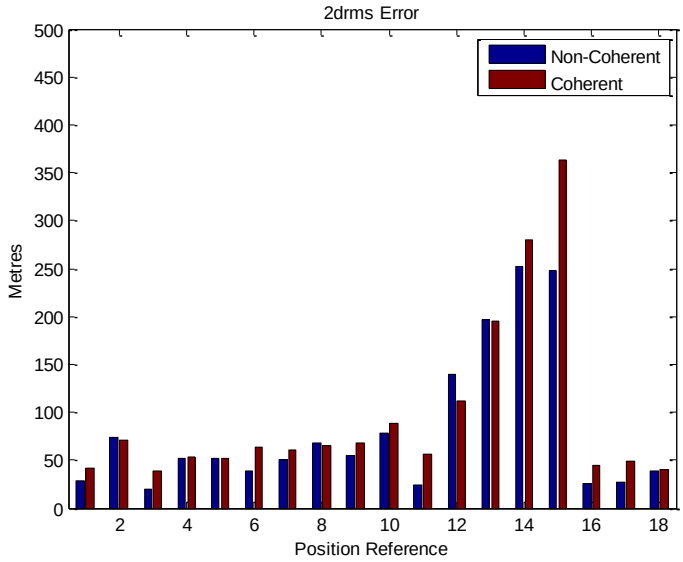


Figure 64. Amazon Static Position Points - 2dRMS Error

Reference points 11-15 show huge error growth. These points are taken from a tag deployment that experienced a data download error that resulted in a corrupt end timestamp. The basic temperature correction for time drift therefore had no end point and relied on the calibration coefficients from the start. As will be discussed in the "Time Correction" chapter, this leads to a gradually increasing time offset which, in turn, leads to an increased position error.

5.8 Extended Snapshot Integration

The research on weak signal detection up to this point used the standard TrackTag snapshot length of 24ms. This was sufficient to demonstrate the performance advantages of coherent integration and the basic functionality of the "All Bit Permutation" technique developed to enable long coherent integration. However, the performance of the technique could only be realised with integrations spanning multiple GPS data bits.

5.8.1 Extended Integration Algorithm Development

A 76ms snapshot known to have a strong signal was used to produce the Phase plots shown in Figure 65. The top plot shows the raw measurement acquired by taking the ratio of I and Q correlator outputs for each millisecond. The 2π measurement roll-over and π shift due to a bit transition were then removed in order to produce the observed phase drift shown in the lower plot. This level of phase drift is intolerable when trying to decipher BPSK. Also, the "All-Bit-Permutation" technique discussed in this paper will not work as planned due to the phase varying even without any bit transitions.

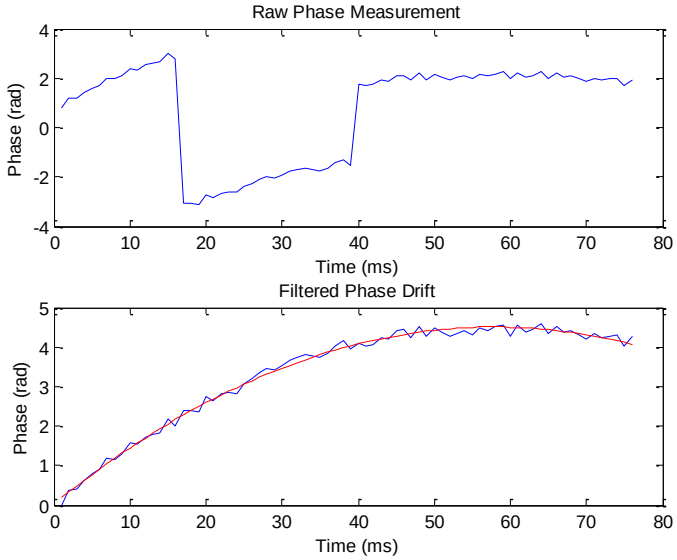


Figure 65. Raw and Filtered Phase (Measurement acquired by taking ratio of I & Q correlator outputs)

The method for eliminating the phase drift is to firstly measure the drift and then apply an appropriate level of inverse phase on each millisecond in order to cancel-out the drift. This has to be done on the signal on at the input to the correlation process as it is re-calculated. Of course, if the system holds the I and Q outputs for every millisecond

for every signal at every Doppler Bin, the correction could be applied to those I and Q values.

The inverse phase is therefore stored as a LUT (Look-Up-Table) to be applied to the subsequent signals. To show that this is a drift that is common to all satellite signals, 3 other satellite signals were processed using the same Phase LUT. Figure 66 demonstrates that, by applying the inverse of the measured phase from the strongest signal, their resultant phase drifts are linear. The corrected phase for the original signal (used to generate the Phase LUT) is seen as the flat line. The four lines show significant noise content but that is expected for measurement points based on the correlation of only 1ms of data. The original, strongest, signal is seen as the one being mainly flat, whereas the other 3 signals exhibit a slope. The slope is due to their estimated frequency being slightly in error. The slope at this point can be used to improve the frequency estimate using the fact that phase is the integral of frequency.

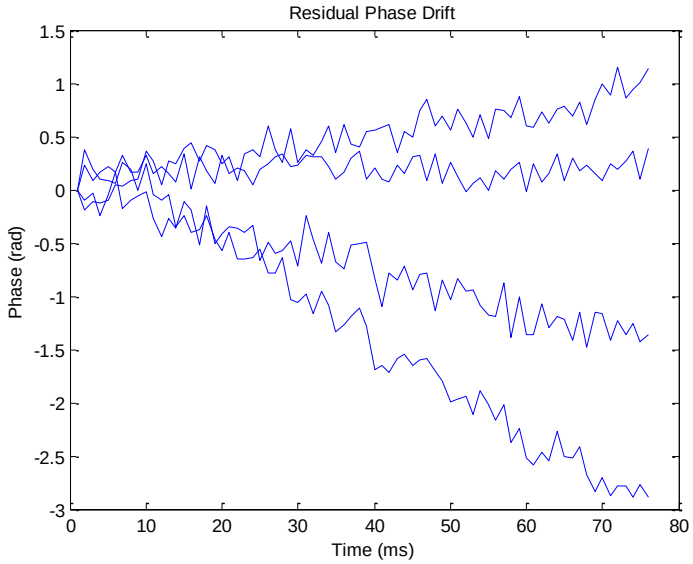


Figure 66. Residual Phase Drift after Inverse Phase Correction applied

Once the frequencies were updated, the 4 signals corrected phase could be plotted in Figure 67. With the phase correction the proposed "All-Bit-Permutation" long integration technique can be utilised. The corrected phase now follows the expected two discrete values with bit transitions 20ms apart in each case.

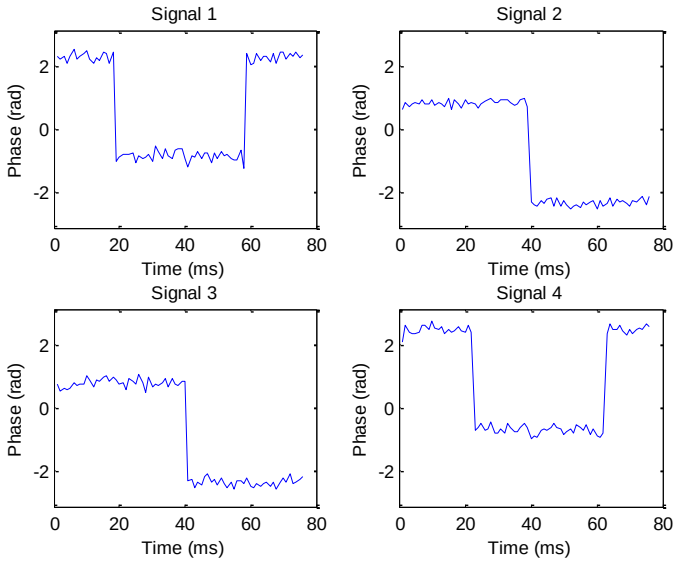


Figure 67. Corrected Phase on Raw Data

25 snapshots, all with strong signals were taken and their correction Phase LUTs calculated for comparison. These are shown in Figure 68. They may appear to be quite different but are, in fact, a very similar parabola only shifted depending on what Doppler Bin offered the highest power. The Doppler Bin with the highest power will be the best compromise frequency over the full snapshot and so, on average, should appear centred, at 37ms in this case. The maximum point on these parabolas indicates at which point within the snapshot the actual frequency matches the frequency estimated by Doppler Bins. In these 25 example snapshots that point is often centred but can be as much as 20 ms either side.

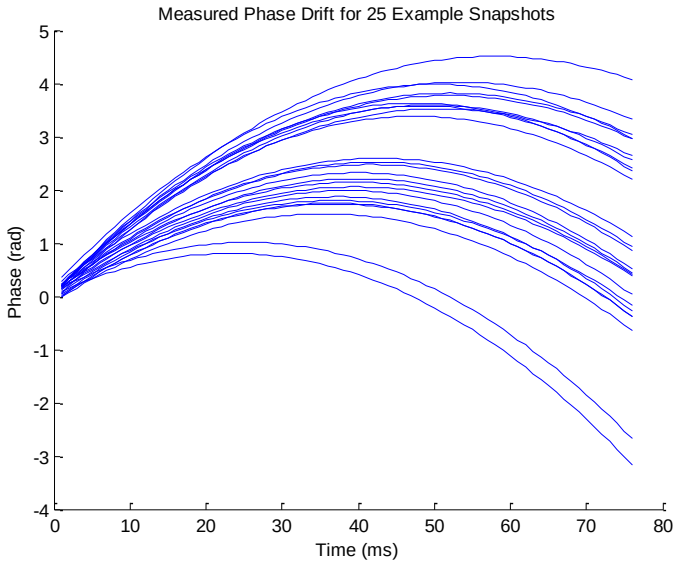


Figure 68. Generated Phase LUT for 25 Snapshots

Figure 69 shows the derivative of the assumed Phase LUTs (i.e. the frequency shift) over 25 snapshots. They are also normalised to start at zero in order to remove the effect of Doppler Bins estimation of the frequency having an error due to the very frequency drift we are looking for. The frequency shifts are all very linear and of approximately the same magnitude. It shows that the phase drift can be attributed to a small shift in frequency over the 76ms. As discussed previously, this is due to the oscillator not having enough time to stabilise properly before the signal is recorded. It may appear to be a backward method for measuring frequency drift, but due to measurement inaccuracies involved the frequency of each 1 ms cannot be measured with the precision required to observe the 0.2Hz shift over 76ms, i.e. 0.26ppm on the 10MHz oscillator.

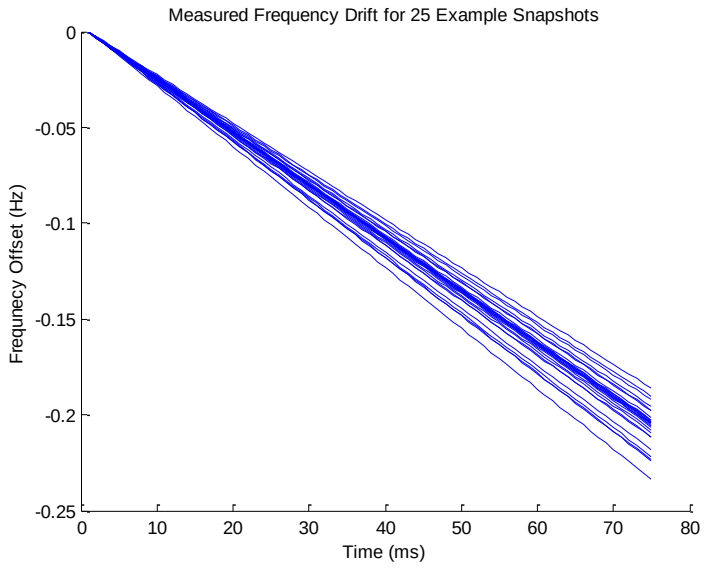


Figure 69. Derivative of Generated Phase LUT for 25 Snapshots

5.8.2 Extended Integration Trials

The TrackTag processing algorithms were adapted to perform 76ms integrations. This involved implementation of the "All-Bit-Permutation" integration technique along with the phase drift mitigation as discussed in this paper. 400 snapshots of data were collected in a forest environment, at a fixed location (surveyed with use of a conventional GPS receiver), in order to compare the performance of the system when using three different integration lengths;

1. 10ms: To replicate common conventional acquisition whereby two blocks of 10ms coherent integrations are calculated and the one giving highest power is used.
2. 20ms: To replicate the "BACIX" technique [67] for coherent integration over one potential data bit transition.
3. 76ms: The maximum possible integration length possible while using TrackTag without making changes to the hardware.

Figure 70 shows how additional signals are acquired through extending the integration length. The total number of signals detected over the 400 snapshots was 1228, 2100 and 3182 for the 10, 20 and 76ms integration lengths respectively. This increase is achieved due to the ability to lower the detection threshold and this is also seen in the plot as the detection threshold falls from 33 to 30dB-Hz between 10 and 20ms integrations. The detection threshold for 76ms integration looks close to the expected 24dB-Hz although some signals at that level may have been discarded due to them being potentially cross-correlation artefact from a strong signal. It is also worth noting that the benefit of moving to an integration length of much more than 76ms may be negligible due to weaker signals being susceptible to cross-correlation products from signals with fairly modest power levels.

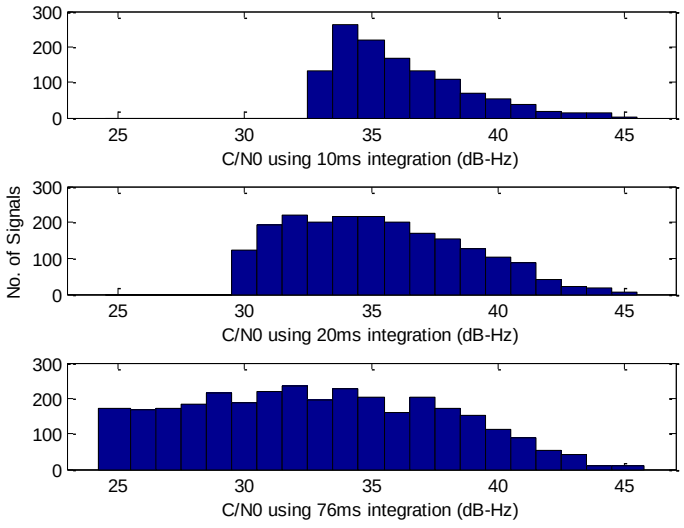


Figure 70. Forest Data - Signal Strength

The result of having far more signals available overall is, of course, an increase in the number of signals detected for individual snapshots. The statistical distribution of the number of signals detected is shown in Figure 71.

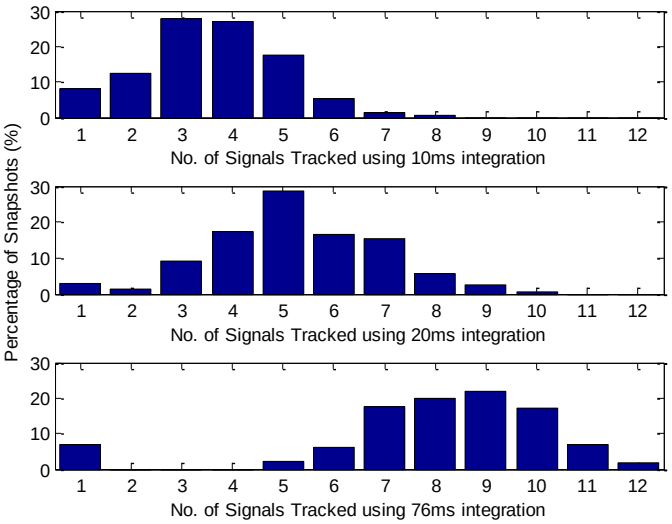


Figure 71. Forest Data - No. Of Signals Detected

The minimal number of observed signals required to make a 3D position fix is 4. However, the position calculation also depends on the DOP (Dilution Of Precision) which is related to the satellite geometry. For example, having the required 4 signals all coming from 4 satellites in close proximity to each other would offer a poor geometry, whereas 4 satellite spread evenly around the sky would offer better geometry and a good "DOP". The more signals that are present, the more likely it will be that the overall DOP is good.

The data presented illustrates that the numbers of observables rises dramatically from an average of around 3.5 to an average of 9 when moving from 10 to 76ms integration. This makes a big difference to the navigation fixes calculated as seen in Figure 72. The overall number of 3D positions calculated is 7, 64 and 317 for the respective integration lengths.

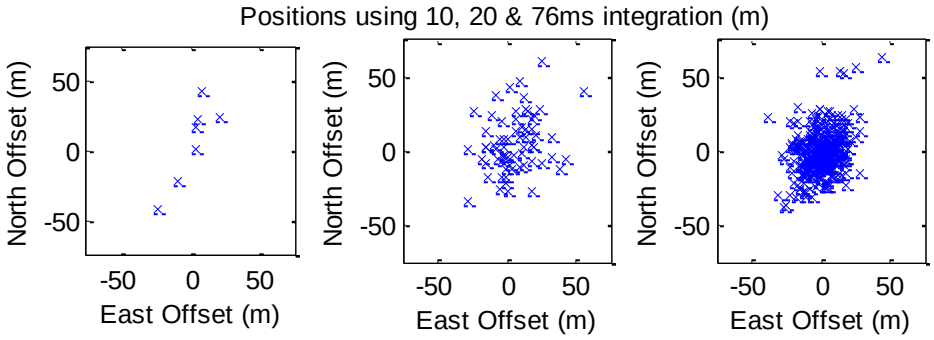


Figure 72. Forest Data - Positions

The accuracy of the positions is also an important factor. Figure 73 shows the distribution of horizontal error for both 3D solutions and 2D (altitude-aided) solutions. It is clear that as well as offering a far higher percentage of successful position calculations, the 76ms integration improves the accuracy of the position fixes. The 2dRMS accuracy of 3D solutions using 76ms was around 29m.

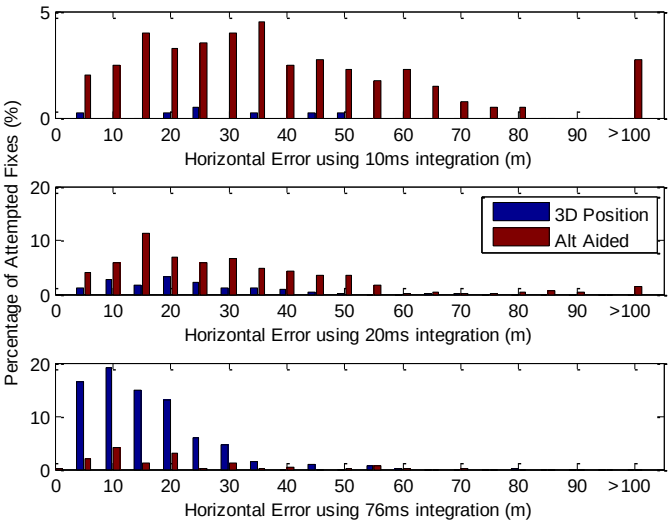


Figure 73. Forest Data - Position Error

5.8.3 Summary

The results have shown that increasing the integration length to 76ms provides significant performance improvement in the ability of the system to provide accurate position fixes in partially obscured environment – a forest in particular. This is of great significance to GPS systems, such as animal tags, where the environments in which the system is likely to be deployed has poor GPS reception.

The exhaustive processing approach to integration of all possible permutations of the received bit pattern and its bit-phase was proven to work. The additional requirement to correct for oscillator drift was necessary in order to make the phase integration work optimally without the need for additional delays to enable the oscillator to stabilise following its "sleep mode". In an application so sensitive to power consumption as long-duration animal tags, everything must be done to minimise the power in order to maximise the battery life.

Following this research TrackTag™ has been successfully deployed in the Amazon and is considered to be the first GPS tag to operate under the Amazon Rainforest canopy.

Chapter 6: Signal Detection Enhancements

This chapter covers developments that were undertaken in parallel with the main body of work.

The Doppler search reduction work was necessary to reduce the processing required for navigation solutions. With the weak signal processing discussed in the previous chapter, the burden of computing over a wide Doppler range made the processing very time-consuming. By minimising the number of Doppler bins to search through, the processing required was also minimised.

WAAS satellites, if visible, offer an additional observable for the navigation algorithms. As much of the TrackTag data was being taken in environments where the number of satellites was severely restricted (in the Amazon Rainforest) the inclusion of WAAS tracking was implemented along with the other archive orbital data available from the internet.

6.1 Doppler Search Reduction

There are 4 sources of error while predicting the Doppler frequency for each signal;

1. Receiver Velocity
2. RF Front-end Oscillator drift.
3. Difference between estimated time & actual time.
4. Difference between estimated position & actual position.

6.1.1 Receiver Velocity

Equation (26) shows that the observed frequency, f', depends on the nominal frequency, f, the observer's (receiver) velocity, v_o, and the velocity of the source (satellite), v_s [69].

$$f' = f\left[\frac{1 \pm \frac{v_o}{C}}{1 \mp \frac{v_s}{C}}\right] \quad \textbf{(26)}$$

To determine the effect of receiver velocity, the satellite velocity is assumed to be zero. The observed frequency can be re-written as (f + Δf) and the equation re-arranged to give equation (27);

$$\Delta f = \frac{v_o f}{C} \tag{27}$$

As C and f are constants, the effect of receiver velocity on the Doppler prediction is simply $\Delta f = 5.26 * v_o$. How this might affect the processing is entirely dependent of how fast the object being tracked can travel. A change in Doppler of 10Hz might be considered significant while attempting coherent integration where the bandwidth can be relatively small depending on the integration length. A velocity of around 2m/s would therefore start to look significant and an assumed maximum receiver velocity should be used to help constrain to processing.

6.1.2 Oscillator drift

The 10MHz oscillator used to clock the RF section on the tag is calibrated against temperature. This removes most of the frequency uncertainty – that due to individual oscillator's centre frequency and also some of the drift attributed to temperature. It does this by making use of the fact that the oscillator will drift over temperature according to a constant curve (Figure 74). The calibration system runs the tag over temperature and then estimates the oscillator offset at each temperature by curve-fitting a polynomial. The accuracy for the calibration system can be estimated by looking at an example.

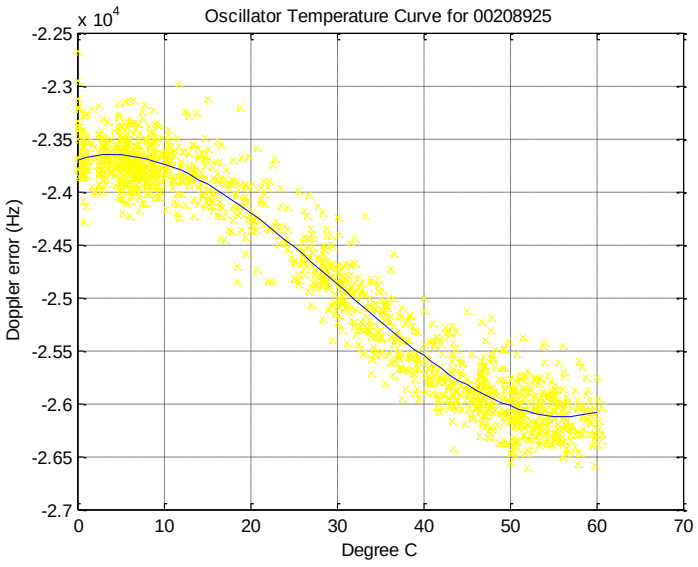


Figure 74. Oscillator Temperature Curve for 00208925

The error after calibration can then be estimated by subtracting the calibration polynomial for the calibration points to get the Doppler error during that cal' (Figure 75). By inspection, the error is within 1kHz of the polynomial. A conservative limit for the Doppler error would therefore be +/-1.5kHz (50% margin above measured data). It is hoped that this error will be further reduced once a more accurate measure of Doppler frequency is made available to the Calibration system itself. At this point the Doppler range of 3kHz is implemented as a configuration parameter (or keyword) in the processing code.

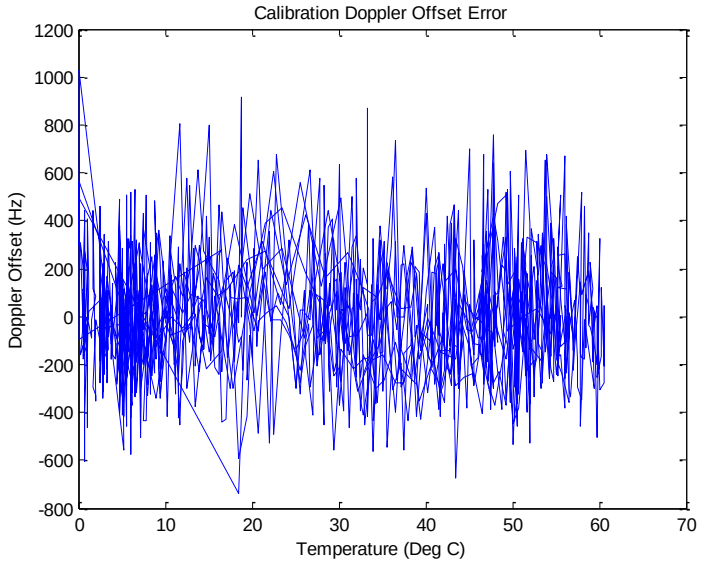


Figure 75. Calibration Doppler Error

6.1.3 Time Error

The effect an error in the Tag real-time clock has on the Doppler frequency prediction is now discussed. This is expected to vary depending on geometry of the satellites. The best way to analyse this therefore is by simulation.

A point on the earth's surface is chosen together with a point in time. The predicted Doppler frequencies are then calculated using a function available within NAVSYS' Matlab toolbox. Time is then offset, backward and forward, and at each point the difference between the new Doppler frequency vectors for all visible satellites is calculated.

The worst case scenario in terms of rate of change for satellite Doppler was assumed to be when a satellite is passing overhead. This is due to the rapid change in relative velocity between the satellite and the receiver. This effect is best illustrated when you hear the tone of a siren change quickly as an ambulance passes by. The change of tone is greatest at the point where the vehicle passes by. For the reason, the point in time selected coincided with the point where the highest elevation satellite was at its peak.

When simulated the assumption that the highest elevation satellite would experience the largest Doppler shift was proven incorrect (Figure 76). The plot shows the highest satellite in red. This satellite reaches approximately 90° elevation but there are two satellites that show a higher Doppler Error.

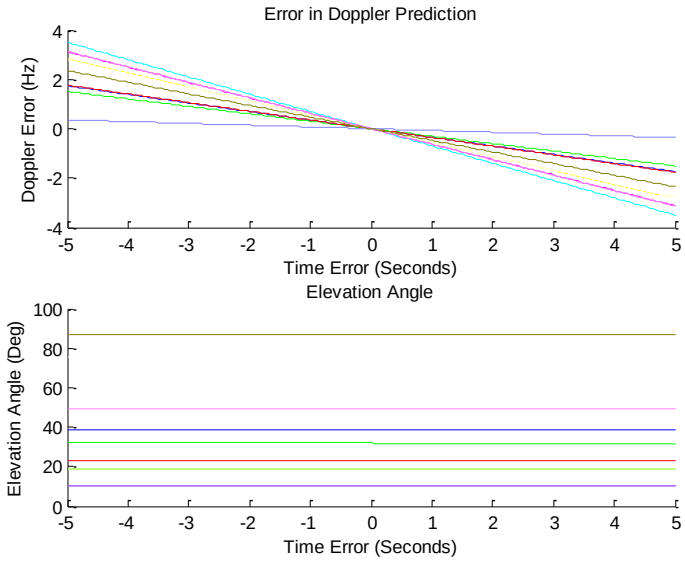


Figure 76. Error in Doppler prediction due to time offset

As Doppler is proportional to velocity, a rate-of-change of Doppler should be proportional to the line-of-sight acceleration. This acceleration is therefore calculated (Figure 77) and shows that the Doppler error due to a time offset is indeed related to the satellite's acceleration relative to the receiver.

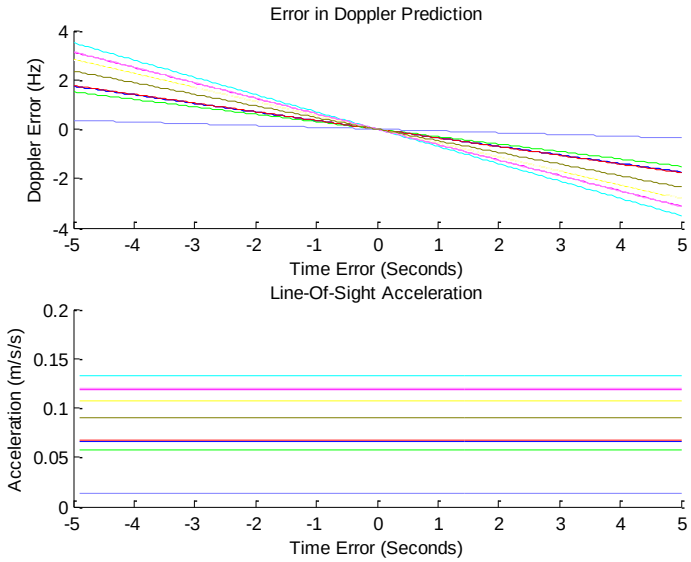


Figure 77. Error in Doppler prediction with satellite acceleration

The implication of this is that the Doppler error due to an error in time is not easy to predict as it depends on constellation geometry at that time. The magnitude of the error will only be an issue for the tracking loops if an attempt to process data with a significantly higher time error (say, > 5 seconds) as the Doppler frequency will then fall into another Doppler search bin in the case of the narrow coherent integration.

6.1.4 Position Error

The effect an error in the assumed position has on the Doppler frequency prediction is now discussed. The same point on Earth that was used in the previous section is now used as the centre of a number of points all of which have their predicted Doppler values compared against the centre values. The points are all on a surface tangential to the Earth's surface. Figure 78 shows the resulting Doppler error. As it was known that the satellite used was at an azimuth angle of 67° (slightly north of east), it can be seen that the largest error is in the direction of the satellite itself. This would appear to make sense as any movement made normal to the satellite will experience very little change in relative distance purely due to the geometry of how much distance would have to be covered to impact the line-of-sight vector. Movement toward or away from the satellite will affect the range. Hence, it would appear that the worst case scenario for Doppler error is movement in the direction of the satellite (positive or negative).

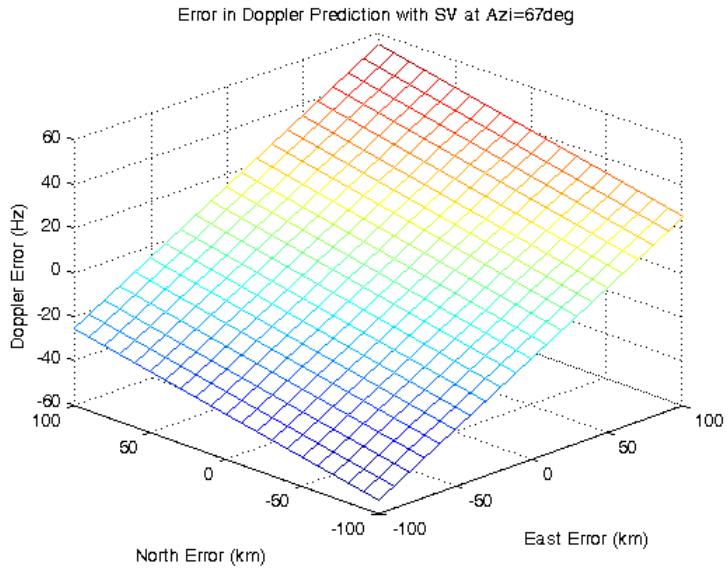


Figure 78. Error in Doppler prediction due to position error

To test this hypothesis, the direction of the Doppler error surface was calculated to see if it corresponded to the azimuth angle of the satellites. As the surface is pretty much linear, this was done by comparing the relative gradients in the north and east directions on the surface. Figure 79 shows that the assumption that the Doppler would change most with movement in the direction of the satellite is not correct. Although the first couple of satellites taken show a close correlation between the two directions, the other satellites show huge differences and highlight the fact that azimuth angle is, in fact, not linked at all to the worst case Doppler error due to positional error.

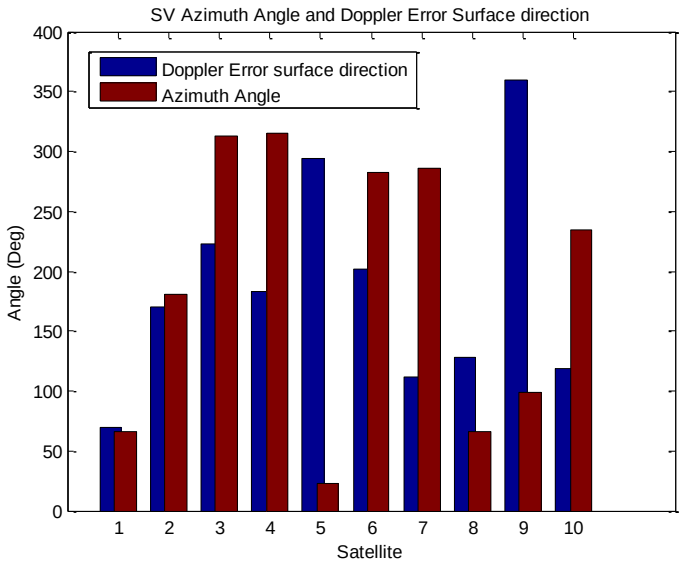


Figure 79. Azimuth Angle and Doppler Error surface direction

Velocity in the line-of-sight of the receiver was then examined. The line-of-sight velocity of each satellite was calculated by taking its position from ephemeris data and its position a fraction of a second later. These two distance vectors could then be used to calculate the relative velocity vector in the line-of-sight. That velocity vector was, in turn, used to estimate the satellite's velocity direction in terms of azimuth. Figure 80 shows the result of comparing the direction based of the Doppler Error surface and that based of the relative satellite velocity. It is clearly a near-perfect match and shows that any estimation of Doppler Error due to the worst case positional error must take the satellite velocity direction into account.

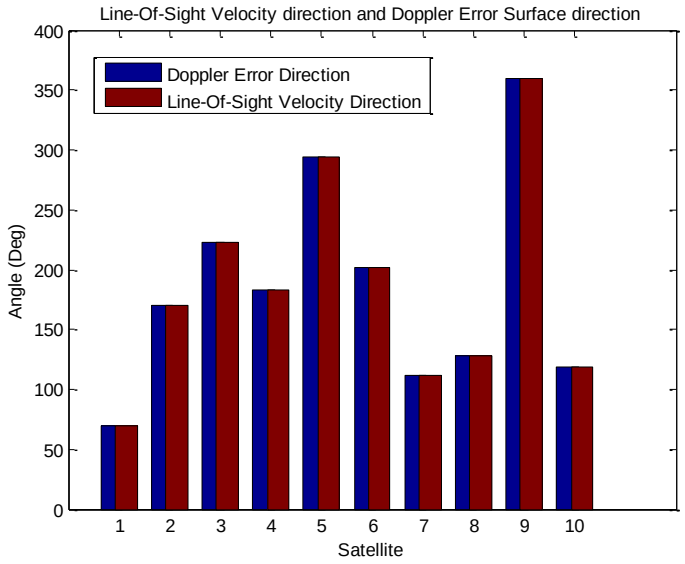


Figure 80. Line-Of-Sight Velocity direction and Doppler Error surface direction

At this point the problem can be reduced from three dimensions to two as only the worst-case is of interest and therefore the Doppler Error for all satellites can be analysed only by taking positional error in the line-of-sight velocity direction thus resulting in a two-dimensional plot. This Doppler error for all 10 satellites is shown in Figure 81, the red line also shown is the Doppler error for movement normal (at 90°) to the velocity direction. The results of taking the angle normal to the direction of velocity all being zero proves that the worst-case angle has indeed been chosen.

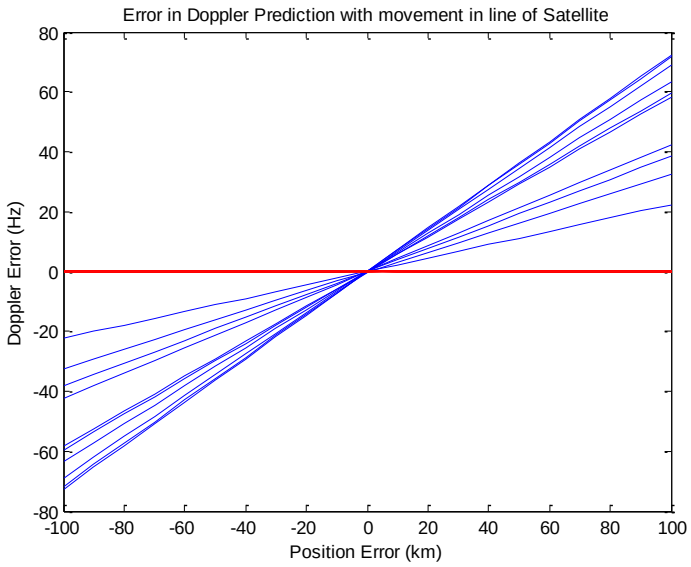


Figure 81. Doppler Error with Position Error in line-of-sight velocity direction

The magnitude of the error in Doppler which has resulted due to the positional error is such that it might cause some problems for the tracking loops.

So, a method for estimating the potential error in the Doppler prediction was required. This was done by assuming the slopes of the Doppler error (Figure 81) are linear and can be approximated simply by taking two points in the satellite's line-of-sight (while maintaining the same altitude) in order to calculate the error gradient. The result of doing this is shown in Figure 82 and demonstrates that the error in Doppler prediction due to horizontal position estimate error can, itself, be predicted.

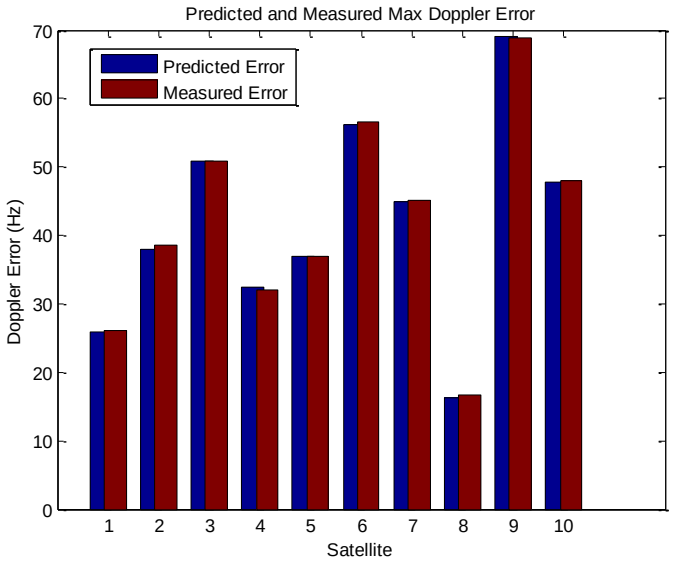


Figure 82. Predicted Versus Measured Maximum Doppler Error

This error prediction can be used when setting the limits for tracking loops in terms of the number of Doppler bins to search.

6.2 WAAS Navigation

6.2.1 Background

TrackTag is intended to be used in harsh environments where it will struggle to acquire enough GPS signals for a position fix. The WAAS (Wide Area Augmentation System) satellites use a similar signal (in analogue terms) as the GPS system. The TrackTag should therefore be able to extract these signals and use them as an additional observable. This requires an innovative approach as the position of the WAAS satellite itself (required to compute a position fix) is usually de-coded from the signal. As TrackTag snapshots are not long enough to allow for the full data de-modulation, another method for determining the satellite position is required.

Figure 83 shows the track of the seven "geostationary" satellites ASTRA over 12 hours and 40 minutes. It is the result of the simple superposition of 368 120-seconds CCD images. The first picture began at 17:11 on January 10th 1998 and the last picture ended at 5:51 the next day. During this time the satellites created these impressive loops visible to the naked eye (North is up, East is left). The image spans 12.8 by 10.5 arcminutes, which means the tracks are spanning around 60km assuming they are at a geo-stationary altitude of 36,000km. It was clear from this that "geo-stationary" is not stationary enough to assume a fixed point for triangulation.

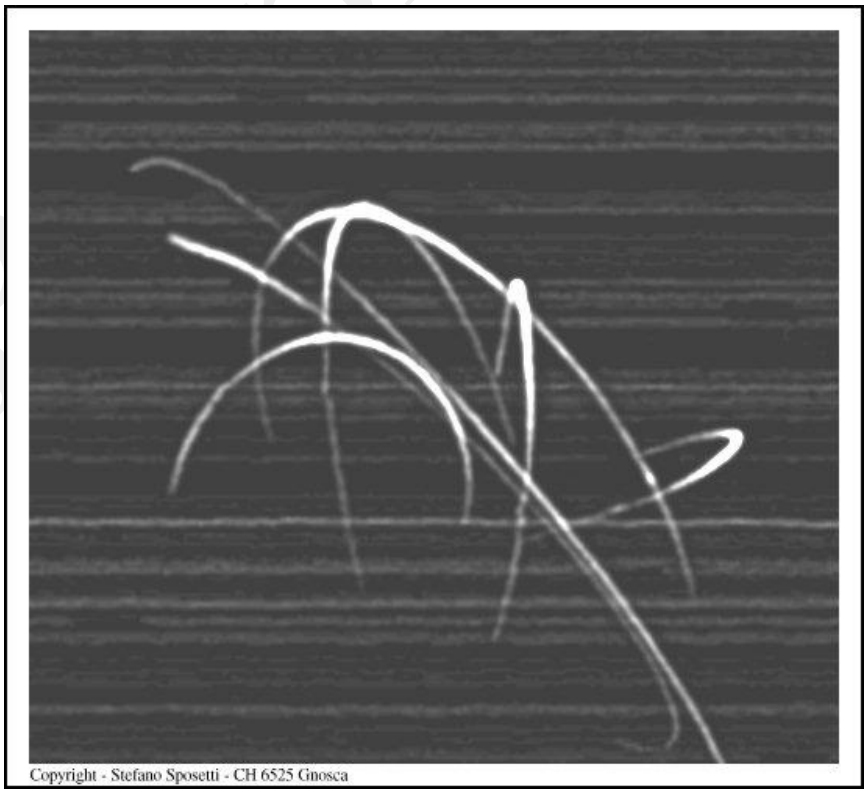


Figure 83. Photo of Geostationary Satellite tracks

Archived WAAS broadcast signals are available on the Internet in Rinex [70] format, a summary of which is detailed in Appendix C. These files were made available during the course of this research and provide full details of what was transmitted by the individual WAAS satellites.

Within all the messages transmitted, and included in these files, is the required information to calculate the position of the WAAS satellite. The data available includes around 337 points per day giving the satellite's 3D position, velocity and acceleration as well as the "Time of Applicability". This data can therefore be used in the TrackTag system to calculate the WAAS satellite's position.

6.2.2 Implementation

As the files are relatively large, file manipulation is very slow. An intermediate file system was therefore implemented such that the orbital data was extracted and saved only once. This data extraction followed the same per-day format as the SBAS files. Translation code was developed to take an SBAS archive and save a file with the relevant GeoNav messages (part of the WAAS signal structure [71]) in Matlab format to file.

The translation code runs from within Matlab and pulls out all the lines of ASCII that are GeoNav messages. Each message is then fed into an embedded Mex C-file in Matlab to decode the message and return the data into a Matlab structure. The "time of Applicability" is the same for some messages as they are repeated 3 times. This redundancy is therefore removed while the messages are decoded.

The need for the intermediate files to improve speed performance was demonstrated by running a test involving the use of 11 days of WAAS data (Figure 84). It was run firstly with no intermediate files available and took 66.6 seconds. As that first execution had generated all necessary files, the second time the code was run it took a mere 0.2 seconds.

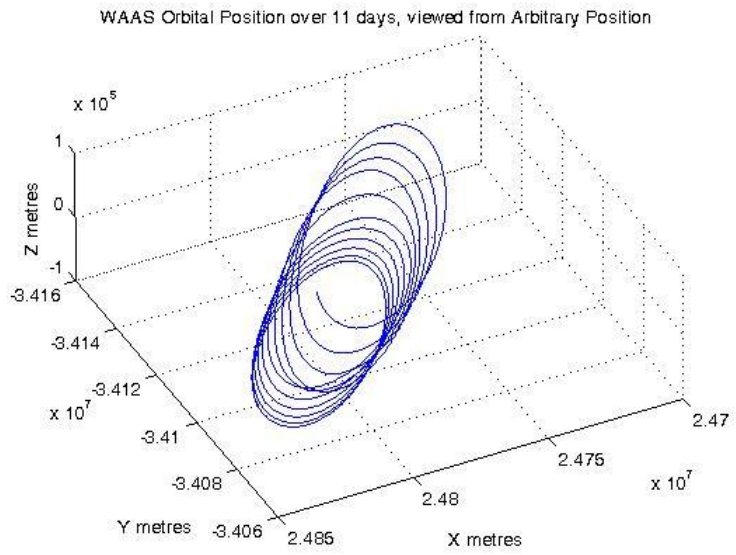


Figure 84. WAAS 3D position over 11 days

A higher-level function was then written to be compatible with the existing main tracking algorithms. The main navigation code therefore looks for the intermediate file, if not present it will look for the full SBAS file and call the translation function to decode (and generate the intermediate file for future use).

Conventional WAAS navigation would use the last position together with the last velocity and acceleration parameters to estimate what the current satellite position is. As there were points both before and after the time of interest, a simple interpolation between the points gave a reasonably accurate estimate for satellite position. This could be done by curve-fitting a polynomial to the position data.

At this point we should also note that the TrackTag data may be very long duration – many months worth. The data shows a mainly daily periodic pattern. It does, however, show some significant changes between days – this is clearly visible between days 6 and 8 in Figure 85. For this reason, it was decided that the WAAS position estimation using curve-fitting should be done on a per-day basis.

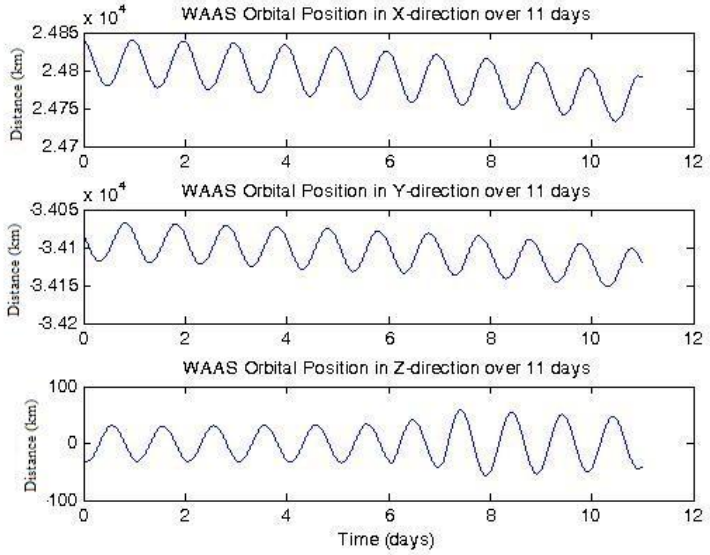


Figure 85. WAAS X, Y and Z position over 11 days

As the tracking algorithm would have to use the daily 337 3D positions and time of day to estimate the exact position at the time of interest (GPS time), some form of interpolation would be involved but this would be more efficient if it were only done once – at the time the data is extracted from the SBAS message archive. The SBAS translation code could therefore modified to generate the polynomials of position versus GPS time (GPS week and GPS seconds) instead of just recording the points taken from the archive.

6.2.3 Approximating Position using Polynomials

The tracking code will request the satellite position for a given GPS week and seconds value. In an effort to reduce runtime complexity the construction of polynomials to approximate the position over time was investigated.

The time recorded in the WAAS message is specified as being < 50ns of GPS time. It is, however, only provided as a second of day measure. The WAAS archive message can also start with a message which corresponds to the previous day. This means we have a direct link between the WAAS time standard and GPS, however, there is a requirement for careful interpretation/filtering of the messages during the extraction of the data.

As the WAAS archive data is provided as daily files, it would seem appropriate to construct polynomials on a daily basis. [Sinusoidal curve fits were also attempted but

performed very poorly compared to high-order polynomials.] Figure 86 shows the 11 day Z position with the polynomial estimate plotted on top. As the polynomials are provided on a daily basis, it is perhaps not surprising that we see some significant errors spaced around a day apart. Similar glitches were observed also on the X and Y position curves.

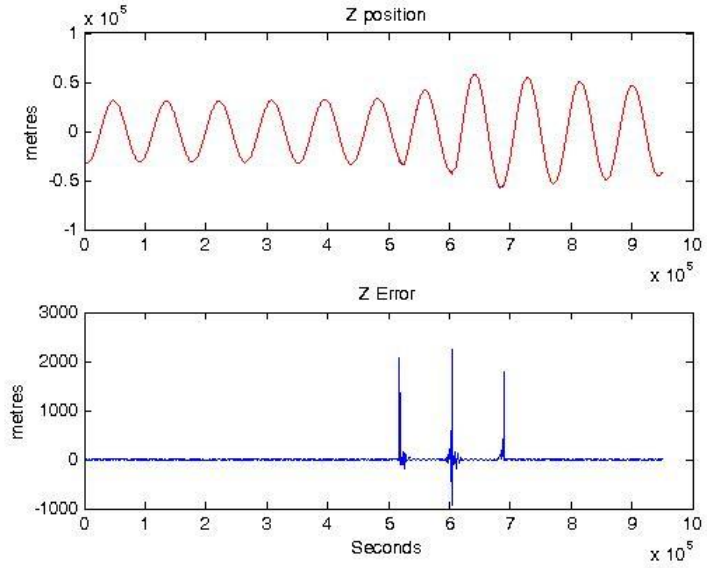


Figure 86. Polynomial Approximation of Z position over 11 days

Figure 87 shows the position and polynomial error over 1 day. The effect of curve-fitting near the endpoints can be clearly seen. It was then thought that perhaps taking points from the neighbouring days could reduce this effect.

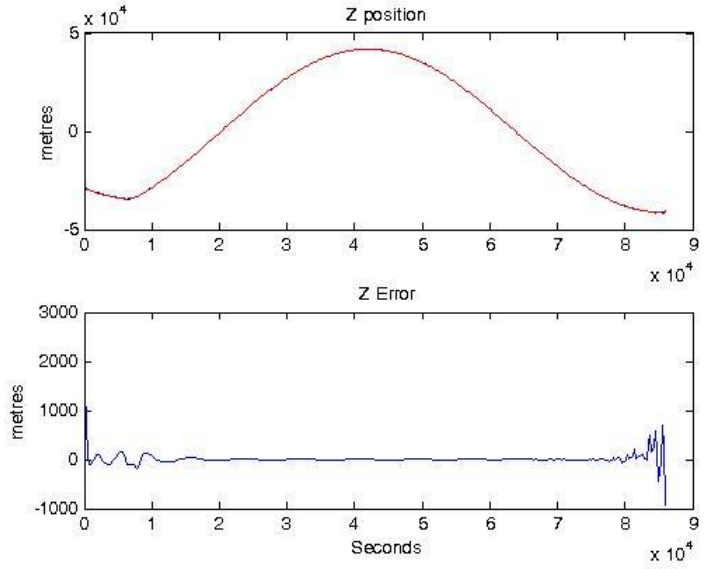


Figure 87. Polynomial over 1 day

By constructing a 30th-order polynomial using all the data from the day in question plus the days before and after, the resulting error is shown in Figure 88. By taking the extra data points, the curve fitting actually has poorer performance – up to approximately 1.5km out! This is probably due to the polynomial having to make more compromises to fit all data points as best as it can.

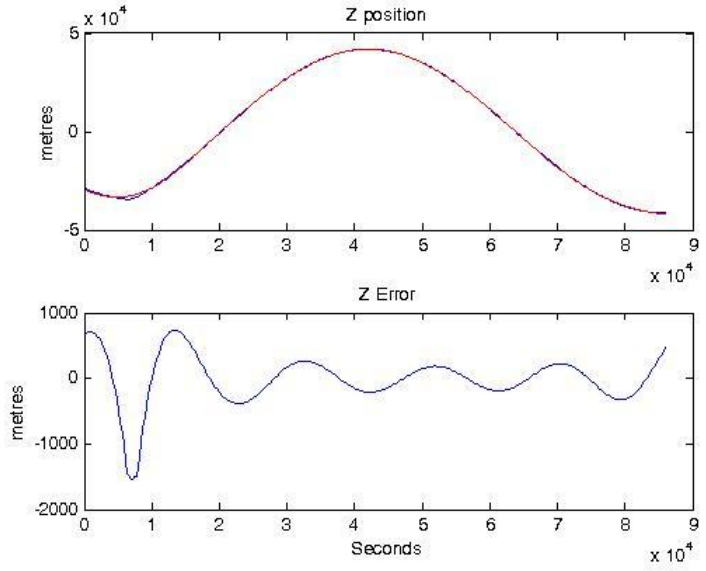


Figure 88. Polynomial Error using Neighbouring day points

This supports the assumption that if polynomial curve fitting is to be used, the data has to be split into small sections. It also suggests that daily sections are too long. Effort was therefore applied to representing the data in terms of hourly polynomials. Hourly sections may give a problem in terms of storage of polynomials over long-duration data sets (many months), but it has be to seen whether that would even reduce the error term to something manageable or whether the polynomial approach is not suitable overall.

Figure 89 shows the result from using hourly 10th-order polynomials. The error has been reduced to around 20m. However, there are now 24 sets of 10th-order polynomials. With there being only around 337 points in the raw data we are trying to fit a curve to, the added complexity and errors induced by curve-fitting makes the process less attractive than first envisaged.

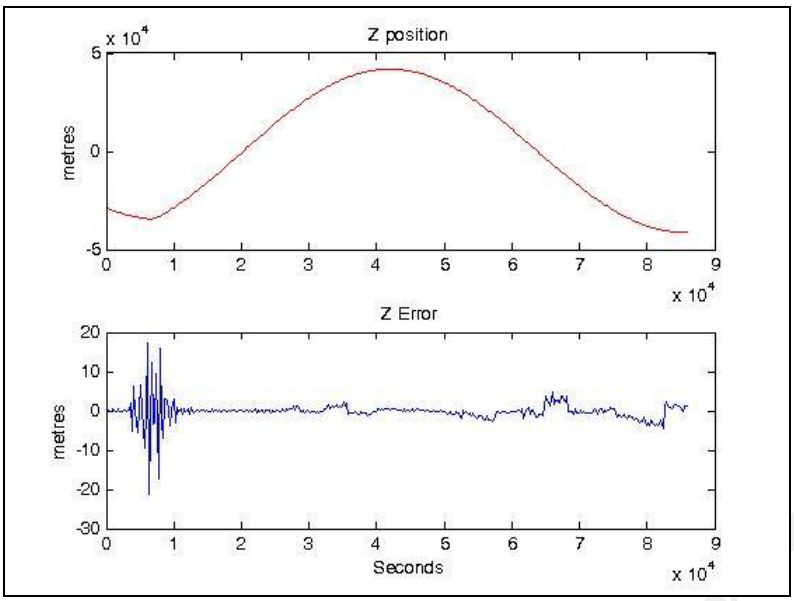


Figure 89. Z Error using hourly polynomials

Figure 90 shows an anomaly found on X position data. There is a discontinuity in the archive data at around $9.39*10^5$ seconds. This will cause the polynomial curve fitting algorithm a major problem when trying to fit a curve to a discontinuity. The cause of this step was found to be an update to the orbital parameters which resulted in a change to the reported satellite clock bias as well as a change to the reported position. This is normal behaviour and is routinely done but obviously causes problems when trying to curve fit to the orbital position.

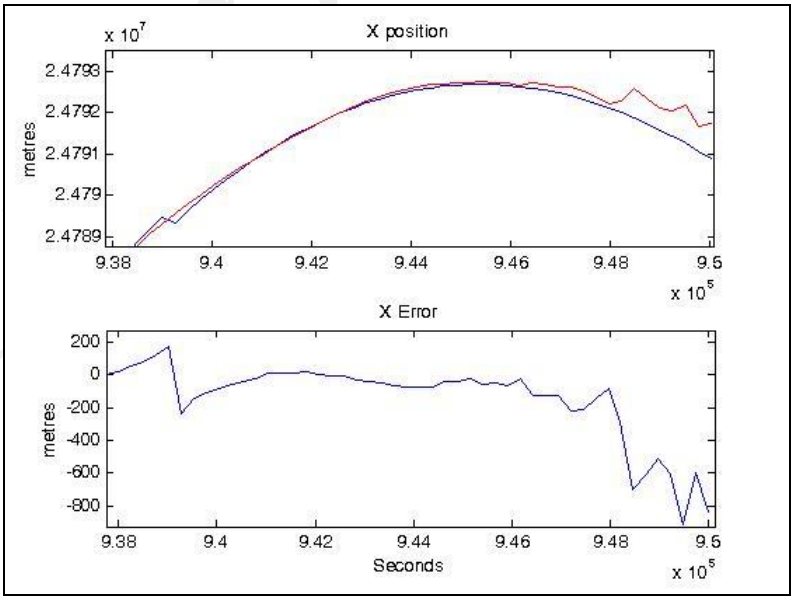


Figure 90. Anomoly in X position data

6.2.4 Interpolation Summary

There are significant issues relating to curve-fitting polynomials to the WAAS position data. It was decided that from this point the raw, time-stamped position data would be stored in an intermediate file. The navigation code would therefore have to read the appropriate messages either side of the time of interest. The estimated position (and satellite time offset) is interpolated between the two.

As well as position, the WAAS messages provide velocity and acceleration (and satellite clock rate-of-change) and these are included in the interpolation. The accuracy field within the WAAS message is also used as a health flag.

6.2.5 WAAS Navigation Results

With the addition of WAAS tracking algorithms and WAAS navigation range estimates (based on the RINEX data taken from the Internet) improvements were seen on the Amazon animal tracking studies.

Figure 91 shows the "NED" (North, East and Down) distance from the starting location for a Tapir living in the Peruvian Amazon. The components of the NED vector are shown in blue, green and red respectively. The navigation algorithms were able to calculate 76 positions using only GPS satellites. This was considered a success due to the fact that no other GPS-based tags had recorded ANY positions in the given environment. [NOTE: The tag spent the first half of the trip time initialised and ready but obscured from the sky while the researchers waited for a Tapir to be caught.]

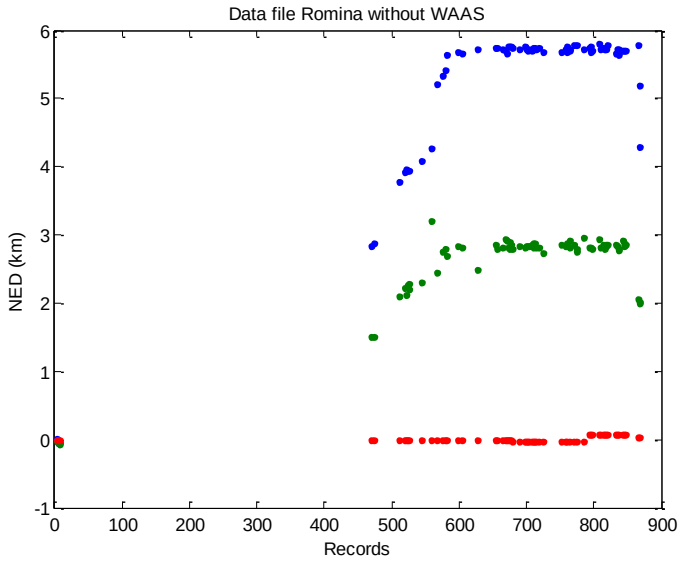


Figure 91. "Romina" without use of additional WAAS signal

Figure 92, however, shows what is achieved when just one extra observable is introduced by tracking a WAAS satellite in addition. The number of positions more than doubled to 183 using the same data. This equated to an availability of over 40% when you consider that the tag was not actually sent outside until around snapshot 450.

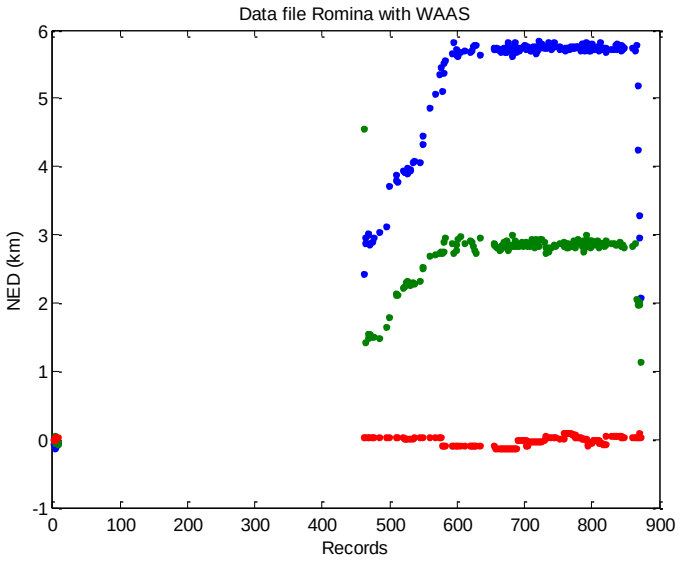


Figure 92. "Romina" with use of additional WAAS signal

Chapter 7: Time Correction

7.1 Motivation

One of the TrackTag product objectives is to offer "long duration" capability. This refers to studies lasting a year or more. Battery and memory capacity issues have largely been addressed by NAVSYS. The major issue remaining is how to maintain an accurate measure of time as the on-tag RTC clock drifts. This time error directly impacts the accuracy of the position fixes and over a year will be unusable. Methods for correcting drift are examined here.

The tags have their RTC set to GPS time at the moment the tag is initialised for a deployment. This GPS timestamp is provided by a conventional receiver integrated with the tag interrogator and is accurate to 1ms. Upon collection and download of the tag, the tag time is compared against GPS time and the drift measured is recorded along with all of the snapshot data.

The Temperature Correction section, for example, was used without any other form of time correction for the data acquired during field trails, conducted by the British Antarctic Survey, tagging Albatrosses on South Georgia between February 3rd and the 24th 2004. Because the precise start and end times are known, the on board clock's accuracy can be measured. In that particular data set there was a time error of 30.87 seconds over the 21 days.

If nothing is done to correct this error the processing of positions will use time with a error of up to 30.87s. What this relates to in terms of error depends entirely on the geometry of the tracked satellites. However, a GPS satellite on the horizon travelling toward the receiver would experience a change in the line-of-sight range of 28.5km in that time! That magnitude of line-of-sight range error is disastrous as any range error will directly affect the accuracy of the computed navigation fix.

7.2 Accuracy Measurement Overview

GPS accuracy is a complex topic involving a variety of technical factors. In an advertisement, a detailed outline of these factors would be, at best, out of place, and at worst, completely meaningless to many readers. The inherent complexity of the topic, coupled with manufacturers desire to show their product in its best light, results in most commercial GPS products advertised specification simply glossing over several important points.

If the same GPS system were to be described by different manufacturers, you would probably end up with varying descriptions. Some manufacturers may use an aggressive style and state the best accuracy that they were able to achieve under optimal conditions (even though such an accuracy may have been achieved only 50 percent of the time). At the other extreme, some manufacturers may be overly conservative. A conservative manufacturer may characterise the receiver under difficult or extreme circumstances, then state an accuracy that reflect the observed results at 95 percent probability. The same receiver, described in two different ways could have two very different accuracy values. For example, imagine that a certain GPS receiver collected 1000 data points under ideal conditions. It is conceivable that the best data point could be accurate to better than a centimetre, and the worst, accurate to only 15 metres. Now imagine that the same receiver collected 1000 data points under difficult GPS conditions (such as under tree-canopy) and under these difficult conditions, it is conceivable that the resulting average accuracy could be degraded substantially.

Expression of accuracy

- CEP (Circular Error Probable) - Values stated as CEP apply to horizontal accuracy only. Half of the data points fall within a circle of this radius centred on truth, half lie outside this circle.
- 1dRMS - Approximately 68 percent of the data points occur within this distance of truth. It should be expressed clearly whether the accuracy value refers only to horizontal or to both horizontal and vertical.
- 2dRMS - Approximately 95 percent of the data points occur with this distance of truth. It should be expressed clearly whether the accuracy value refers only to horizontal or to both horizontal and vertical.

- 3dRMS - Approximately 99.7 percent of the data points occur with this distance of truth. It should be expressed clearly whether the accuracy value refers only to horizontal or to both horizontal and vertical.

Some static data was taken and results analyzed with the aim of quantifying the positional error induced by an error in time. 500 sample snapshots were used and Figure 93 shows the percentage of those that fell within the horizontal distance from the centre point. The time offset applying varied from 0 to 1 second in 0.1s increments and the centre point was taken as being the mean of the positions calculated with no time offset.

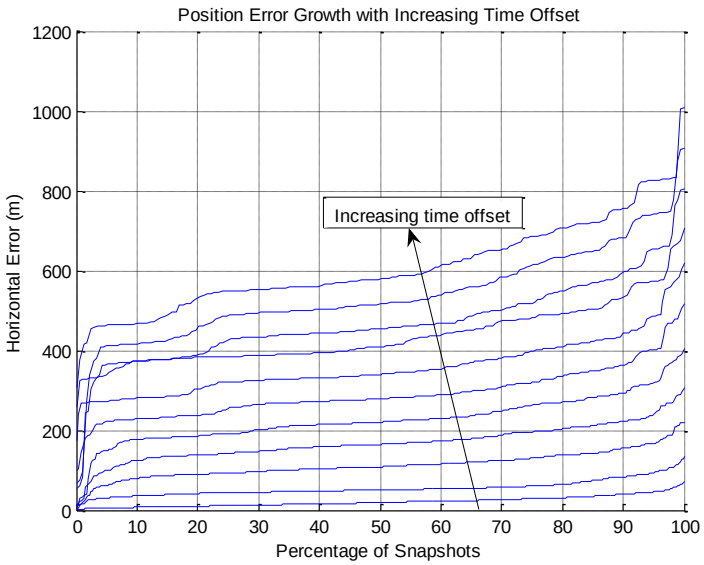


Figure 93. Position Error Growth with Increasing Time Offset

The lowest line, indicating the smallest errors, corresponds to no time offset, the highest line corresponds to the maximum offset of 1s.

It was also observed that offsetting time introduces an overall position drift. This is shown in Figure 94 and explains the sudden jump in error found in Figure 93. It is assumed that this is due to the geometry of the satellites used for the particular snapshot having a bias in terms of the average satellite relative velocity. That hypothesis would suggest that, in this particular case, the average relative velocity of the satellites used would have been in a north-westerly direction. This is completely consistent with the GPS constellation satellite planes and inclination.

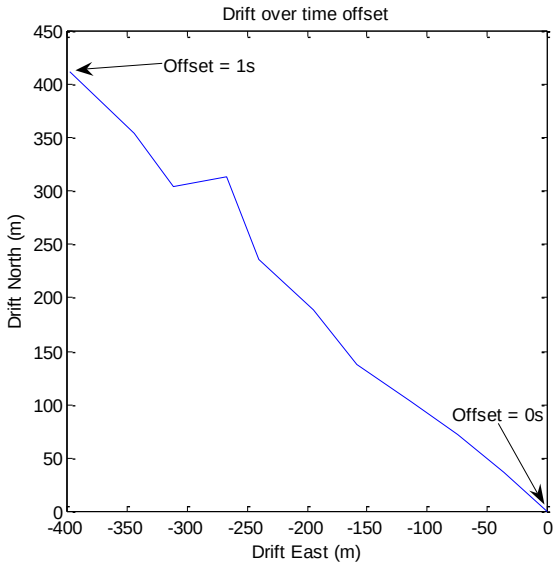


Figure 94. Drift over Time Offset

Expressing this data using the commonly used terms for GPS accuracy gives us the plot shown in Figure 95. It shows a near linear growth in horizontal error against time error. The reason it does not give no error with no time error is due to the signal detection measurement noise and the fact that the assumed time with "no error" will, of course, have an unknown error.

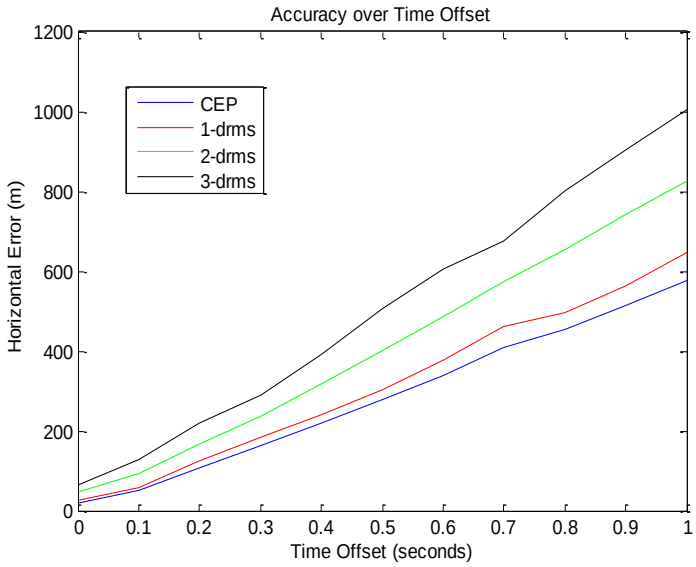


Figure 95. Accuracy over Time Offset

It was been decided that within this thesis (and subsequently within NAVSYS) GPS accuracy will be expressed as 2dRMS, i.e. the error within which 95% of points lie. The target accuracy will vary for customer applications. The graph shows us that for

100m 2dRMS accuracy our time estimate has to be within around 100ms. This provided a ballpark figure for the research to use in terms of investigating what is required. The processing system could be augmented to include a user input that specifies desired accuracy and therefore specifies how accurate the time estimate needs to be.

7.3 Temperature Correction

In an effort to correct the time estimation, there is a temperature sensor on the tag which allows some temperature correction of the local oscillator given that its frequency varies as a parabola against temperature. The parabola could thereafter be used in the post-processing to adjust the recorded time according to the temperature for each snapshot taken. Having performed the temperature correction on the Albatross data there was still an error of 1.196 seconds. This error between measured time and temperature corrected time is then removed by simply applying a linear correction across the entire timescale.

There are two problems when doing this; Although the time at the very start and end points will be accurate, there is no way of knowing what the time error is throughout the data and the temperature correction is assumed to give accurate time estimation. The figures discussed above corresponds to data taken over a period of 21 days, and if the system is proposed to be used for year-long trails, it is clear that another method for time correction is required.

There is a high probability of having a tag retrieved with a flat battery (thus losing the on-board RTC time) and therefore no temperature and time data for an indeterminate length of time. In that case, there would be no way to apply the final linear approximation to give accurate time at the end of the data.

The RTC crystal used on the tag was the Citizen CM130 which has a drift specification of 20ppm at 25°C. As the crystal is supplied uncompensated and so follows the usual parabolic drift centred at approximately 25°C. If nothing is done to improve the effective RTC drift, the maximum drift permissible for the tag design (given in the last section) of 100ms would be crossed in around 83 minutes – clearly unacceptable for the "long-duration tag"!

It was decided that the tags RTC crystal would be calibrated. No attempt was made to hardware tune the RTC crystal on an individual tag basis as it was determined that correction using an estimate of the individual crystal's temperature drift parabola would be required which would have made any 1st-order hardware adjustment redundant.

The calibration had to be done across the operating temperature range due to the parabolic relationship between crystal drift and temperature. A calibration system was constructed that subtly altered the normal operation of the tag in order to measure the RTC clock drift. The design used during calibration had an extra wait state inserted before the tag would timestamp each snapshot. In order to exit the wait state, the code would poll a 1pps signal taken from an external OEM GPS unit which was locked to the GPS time. This process meant that the tag could be programmed to wake every, say, 3 minutes and the actual timestamps would vary slightly according to how much the tag time differed from the assumed "truth", GPS time. After adjusting for any second rollovers in the measurement the data would then provide the clock drift.

Figure 96 shows the outputs of one example calibration run. The tag is placed in an environmental oven that cycles the temperature up and down. The on-board temperature records the tag temperature for each wake. The "Fractional Second" is what remains when the expected cadence of wakes (3 minutes) is removed from the recorded wake times. The introduction of a wait until a GPS 1pps occurs before the timestamp is recorded means that the fractional second is the amount the RTC has drifted. The drift data can therefore also be related to temperature.

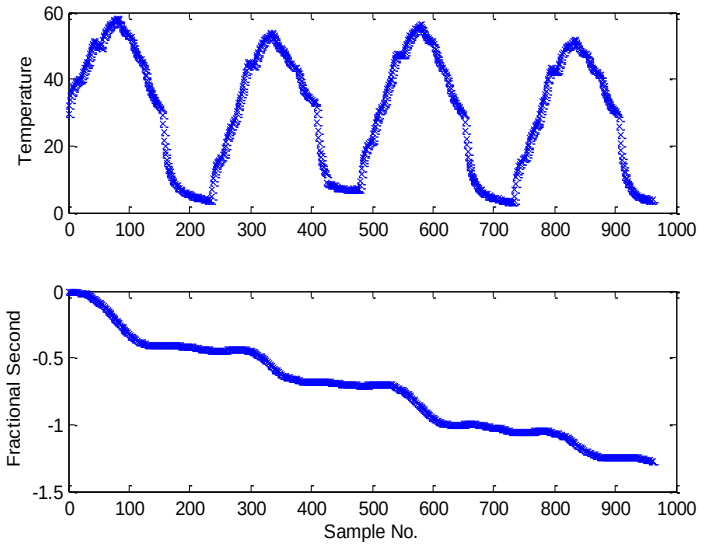


Figure 96. Tag Calibration Temperature and Fractional Seconds

When the data is re-organised and plotted against time (Figure 97) the expected parabolic relationship is clear. The quantisation of the drift shown is due to the RTC reported time itself being quantised by 10ms steps. The temperature calibration fits a

parabola to the data (shown in red) and the parabola coefficients are stored on the tag for use in the processing algorithms later.

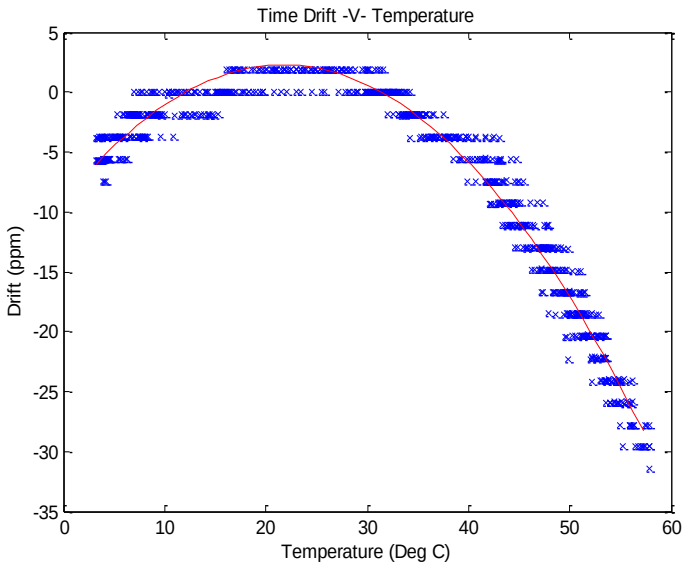


Figure 97. Time Drift against Temperature in Calibration

Of course, there is still some drift error remaining after this calibration, albeit far less than the part specification of 20ppm @ 25°C. The temperature calibration serves to characterise the part-to-part variance in;

1. The parabola turning point, i.e. part drift at 25°C.
2. The sensitivity to temperature.

Following calibration, each tag's time drift can be treated as being of a similar magnitude. This obviously requires the implementation of correction factors using the individual tag's stored calibration coefficients during the post-processing stage. The next step was therefore to estimate how accurate the tag's RTC was after this correction. Calibration data from a set of 10 tags was used to create the distribution plot shown in Figure 98. It shows that there is a Gaussian distribution of post-calibration drift. The 3dRMS point is well within 4ppm while 50% of the error can be found within 0.6ppm.

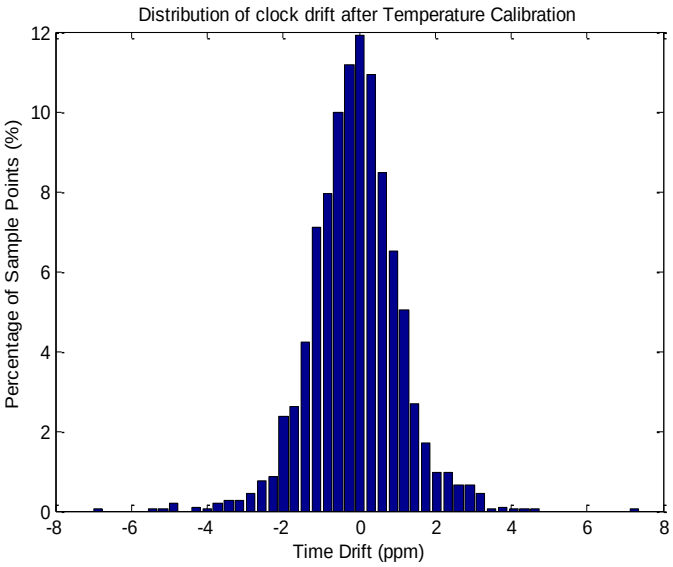


Figure 98. Distribution of RTC Drift after Temperature Calibration

This error distribution is important to the performance of TrackTag as it determines what accuracy to expect the RTC to "free-wheel" at without any additional correction. The 50% point of 0.6ppm would result in the tag being able to run for 46 hours until time error failed whereas the 3dRMS point would allow for only 7 hours.

A further source of error inherent to this process is the fact that the actual temperature will not rise/fall between snapshots in the near-linear manner achieved during the calibration run. The variability of temperature between snapshots will be random and would tend to get worst with longer snapshot cadence. For course, taking temperature points at a rate far faster than temperature could drift would be a way to reduce this error but this was not practical for this application.

7.4 Endpoint Adjustment

As TrackTag is downloaded at the end of a deployment, the GPS time (and position) from the TrackTag Interrogator is stored in the same download file. Crucially, the difference between the tag RTC time and GPS time is also stored. This could therefore be used to correct any drift experienced by the tag's RTC. The post-processing algorithms therefore applied the temperature calibration adjustments to calculate the temperature corrected time at that endpoint which could then be compared against the actual time to measure the remaining "post-calibration" time drift. The inverse of that drift could then be linearly applied across the entire dataset resulting in an improvement to the effective RTC accuracy overall. This only works well when the download (and corresponding measure of RTC drift) takes place at a time close to the time when the last snapshot was taken. This, for example, would not be the case if the tag had filled its snapshot memory long before being collected.

It is very difficult to estimate how much improvement is experienced through use of the endpoint time correction. It is very much dependant on how the clock drifted in reality during the course of its deployment. The drift is not likely to be totally random but with Gaussian spread as the temperature correction section may suggest. It is more likely that the drift varies slowly and perhaps with some hysteresis with temperature.

This assumption was based in real-world performance of TrackTag. Deployments early on in this research involved tracking Albatross over a period of up to 25 days using only temperature correction coupled with endpoint adjustment. It was evident that the navigation points near the centre of the trip were close to unacceptable due to their accuracy degrading. The middle of the data was more prone to time drift due to the start and endpoints being constrained by separate GPS time-stamping while the points between relied on the time correction.

The actual time drifts experienced by TrackTag before endpoint adjustment over a spread of 10 deployment datasets varied from 0.37 to 1.15ppm. The time drift estimates given in the temperature correction section can therefore be considered "worst-case" and extremely conservative. It was, however, recognised that something else was required to push TrackTag performance beyond 1 month.

7.5 Reduction of residuals

7.5.1 Background

Fundamental to the performance of a GPS receiver is the knowledge of time. To be able to compute a position fix based on triangulation of satellite signals, the time of transmission-to-reception equates to range. Therefore any error in time results in an error in range.

There are two main parts to computing the position, tracking the signal and then calculating of the navigation solution. "Tracking" has been discussed at length through this thesis. However, it is worth noting that before any signal tracking is attempted, it is necessary to estimate which satellites should be in view prior to spending time searching for them over the possible Doppler range. Assuming the position is known to within a few hundred kilometres, the ephemeris data can be used to determine which satellites are likely to be in view. The system developed for this work then attempts to lock on to each of these satellites using their standard PRN codes.

Any satellite signals that are acquired allow the system to calculate the code phase, modulo 1ms which equates to modulo 300km in satellite range which is the input to the next phase - Navigation Solution.

When calculating a position fix the algorithms use the code phases from the tracking loops, with ephemeris data and time to determine "pseudo ranges" modulo 1ms (or 300km) to the satellites tracked. These pseudo ranges are then used, together with the last known position, to triangulate a position fix using a least-squares approach (this assumes the receiver has not moved so far away that it has jumped into another period of the PRN sequence).

The solution normally requires at least 4 satellite signals to resolve the 4 unknowns: Latitude, Longitude, Altitude and time. However, if there is an over-determined solution, i.e. more than 4 satellites signals are tracked, there can be a residual error due to errors in the pseudo-ranges making it impossible for the ranges to meet at the same point. The least-squares approach reduces to error to a minimum.

Actual movement of the GPS receiver will also give a residual error as the position is calculated using the last known position. For this reason, the navigation solution is iterated and completes when the residuals have been minimised.

7.5.2 Hypothesis

The idea behind the Residuals Reduction Technique (RRT) was that the residual error could be further minimised by varying a time offset in the least-squares equations. The required time offset to minimise the residual error would then be equivalent to the time error. As the data is processed, for any over-determined solution, time could be offset while the residuals are calculated thus giving a curve of residual navigation error against time offset.

For any one position where the time estimation is accurate, an over-determined solution would give a small residual error. Whereas any position fixes where time has an error would yield a higher residual error. The relationship between residual error and time offset could therefore be used to adjust the time and improve overall accuracy.

7.5.3 Algorithm Development

Selecting Snapshots for Analysis

As the proposed algorithm looks at the over-determined residual error, a sample dataset was filtered for snapshots with more than 4 tracked satellites. (The dataset used was actually one of the early Albatross datasets taken around the start of this research). Figure 99 shows that there is a good spread of snapshots with an over-determined solution (> 4 tracked SVs) and so there was little restriction on where the algorithm could be tested.

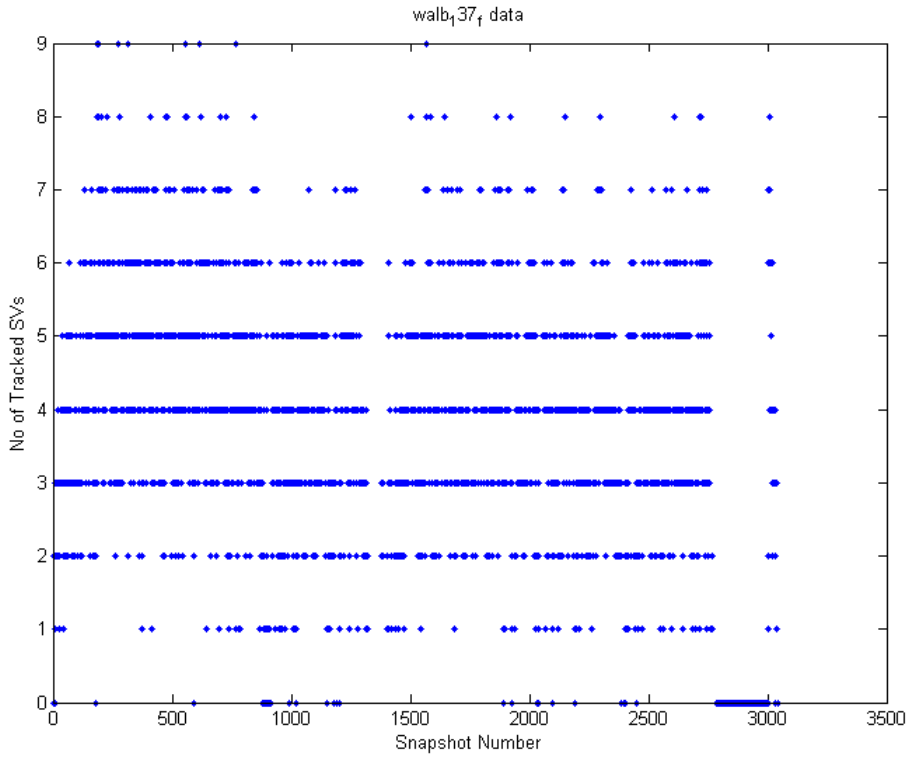


Figure 99. Satellite Count

Snapshot 39 (out of 3042) was the first to have more than 4 tracked satellites (it had 5 SVs). As the snapshot duty cycle was 10 minutes in this case, snapshot 39 was taken 6 ½ hours after initialisation. The time error is therefore expected to be very small.

Snapshot 3019 was the last one to have more than 4 tracked satellites (it had 6 SVs). As this was so close to the end, the actual RTC time error was expected to be close to the measured error after temperature correction, 1.196s.

Snapshot 672 was also chosen for analysis using the residuals reduction technique as it was the first snapshot with significant residual error.

The snapshots had already been processed using only the temperature correction technique and therefore position fixes had been calculated for each. An new algorithm was then implemented to use the estimated position, track the SV signals and then run the navigation loops again while varying the input time offset. The resultant residual error was recorded for each time offset. By plotting this it was anticipated that the true time error would be indicated by a minimum point on the curve.

Initial Results - Snapshot 39

The original residual error was 20.2m which is matched at offset = 0 during this time sweep. However, as Figure 100 shows, even this close to the start of the data, a time error is indicated by the fact that a lower minimum is achievable slightly off-centre. A residual error of less than 10m would have been achieved had the time been 60ms delayed.

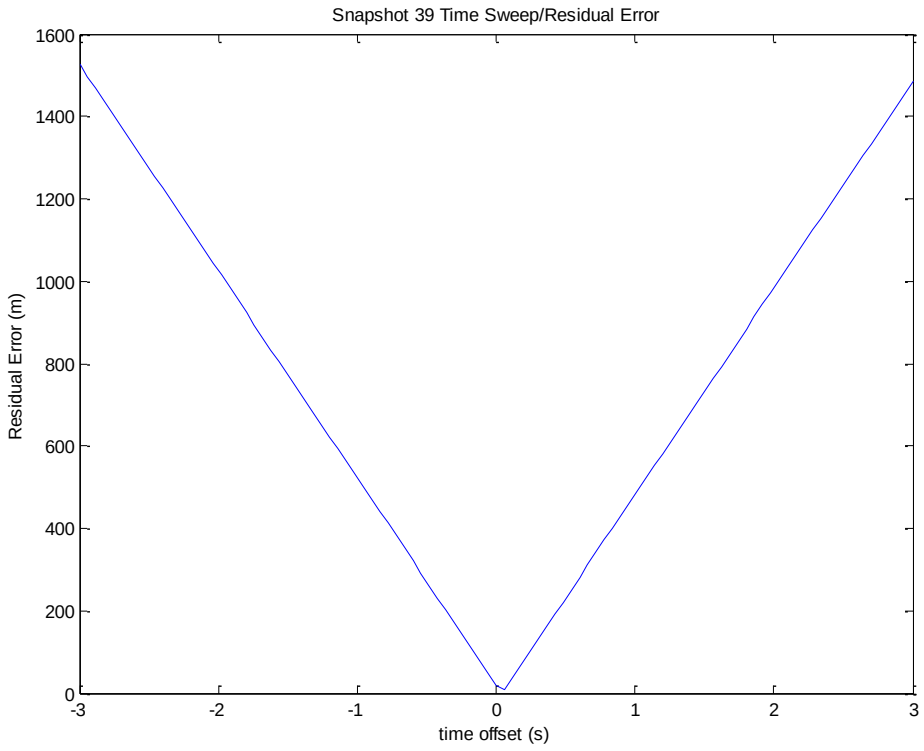


Figure 100. Residual Error over time for snapshot 39

Initial Results - Snapshot 3019

The original error was 77.8m and is matched at time offset = 0 on the graph. Again, as Figure 101 shows, even this close to the end of the data, where time is expected to be accurate, a time error is indicated by the fact that the minimum is slightly off-centre. A residual error of just less than 30m would have been achieved had the time been 120ms delayed.

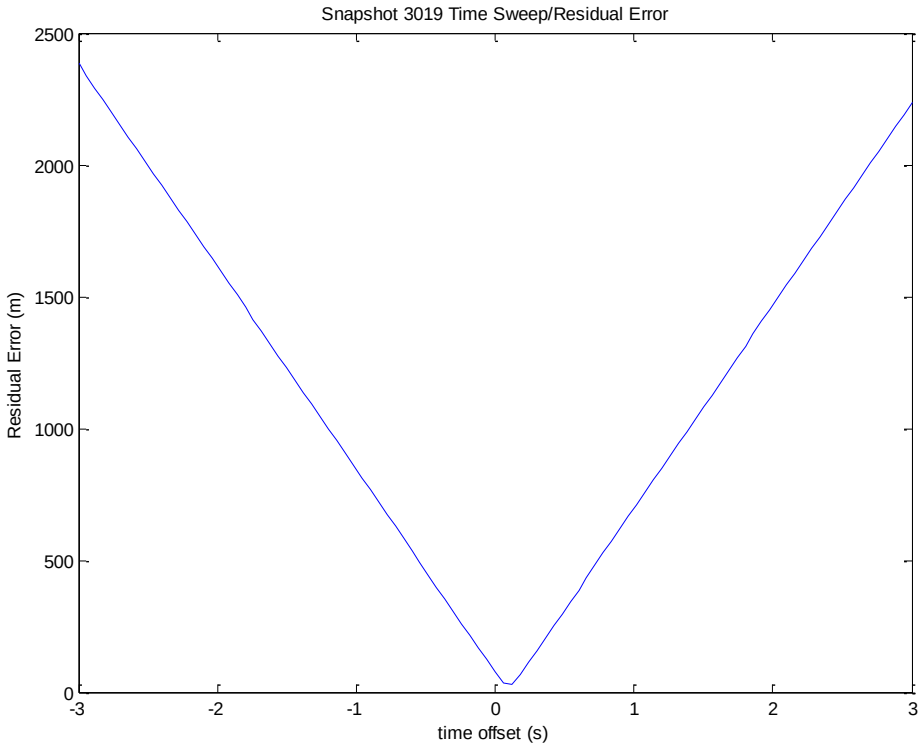


Figure 101. Residual Error over time for snapshot 3019

Initial Results - Snapshot 672

Figure 102 shows the residual error for snapshot 672, although reducing, does not fall to an acceptable level - even though the time offset is +/-50s which is beyond any expected time error as the overall uncorrected error was only 30s. This suggests that there is a problem with the algorithm here.

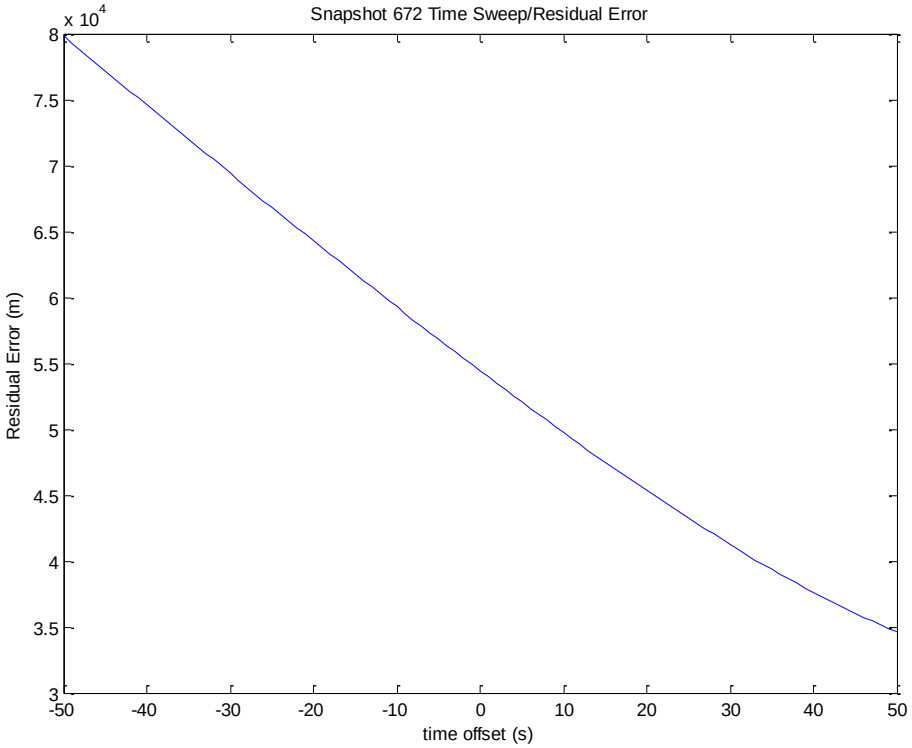


Figure 102. Residual Error over time for snapshot 672

Discussion

The first snapshot taken, although very close to the expected time offset of near zero, indicated an offset of 60ms which meant the drift was a lot higher than expected, i.e. 2.6ppm compared to the 1.3ppm estimate derived in the temperature correction section. The same issue with far greater estimated drift than expected was seen with snapshot 3019.

The failure of the algorithm to converge on a minimum point for snapshot 672 was of greater concern. In order to analyse how the algorithm performs over a sufficient amount of data, the process of searching for the optimum time offset had to be automated. The manually acquired results (only 3 of which are described above but many more were taken) were deemed encouraging enough to merit the effort involved in automation of the RRT technique.

7.6 Residual Reduction Technique (RRT) and Temperature Correction Results

7.6.1 Raw Comparison

Following the implementation of a RRT based time error estimation data was processed with only temperature correction enabled and then with only the residual reduction technique enabled. Figure 103 shows the difference in the estimated time offset between the two techniques.

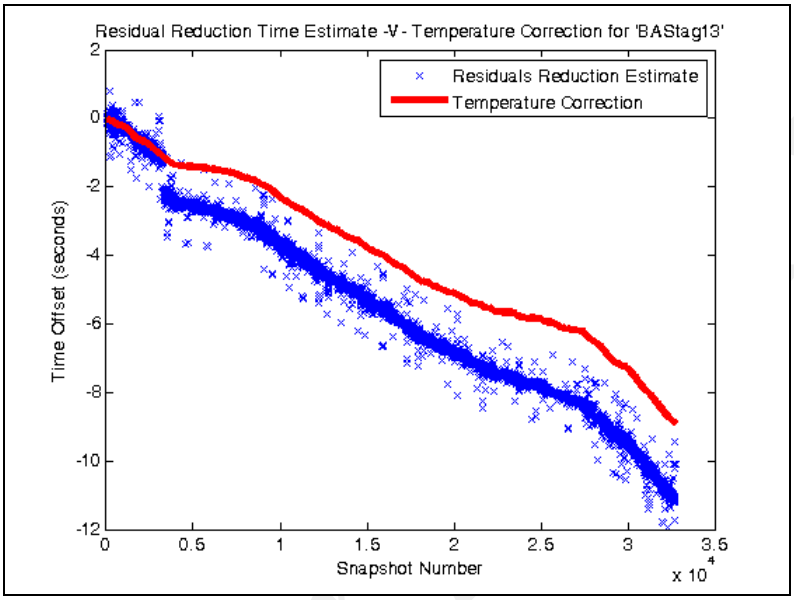


Figure 103. Residuals Reduction Time Estimation Versus Temperature Correction

As the RRT used the real-world data at the time in question to estimate time, it assumed nothing other than an arbitrarily defined maximum permitted deviation of 1 second. It was, however, affected by measurement noise on the pseudo-ranges and can often hit the limit of its search routine – this can be seen in the outliers around 1s away from the average as the RRT search limit was set to around 1 second. At this point where many snapshot's estimated time offsets were plotted it was clear that there was a significant element of noise inherent in the RRT process.

There is also a step change of 1 second near the start of the data. This was found to occur as time rolled on to January 1st 2006. There was, in fact, a UTC leap second at this point that provided a 1 second jump in GPS time compared to UTC. The reason it resulted in a step on the RRT estimate is because the downloaded tag time is assumed to be in the UTC time domain, which it is as that is how it is initialised. However, the tag's RTC knows nothing about UTC leap seconds, so when the change to the leap

second value occurred mid-way through the dataset the processing algorithms applied the step change to a time vector that did not apply it. This issue was resolved in the algorithms but highlights the fact that the RRT will follow the correct GPS time, albeit with some noise. Using the temperature correction alone would not have corrected for that jump.

The temperature correction estimate is based on a drift versus temperature curve that would have been measured months before the data was taken. During that time the clock crystal will have aged and the curve may have changed slightly. Also, the temperature points used are taken as the midpoint between the snapshot temperature and the previous snapshot's temperature. This assumes that temperature varies linearly between snapshots as discussed previously.

So, we are basically comparing a noisy but current estimate of time (RRT) against a smooth estimate which will have a slight drift due to aging and cumulative error.

7.6.2 Temperature Correction and RRT combination

In an effort to combine the long-term average accuracy of the RRT and the short-term accuracy of the temperature correction, the difference between them was plotted - Figure 104. Simple filtering was applied to reduce the noise on the RRT estimate. The graph shows that the difference rises to around 1.3s. Figure 95, in the section 7.2 suggests this would result in a horizontal error of around 1km – clearly unacceptable!

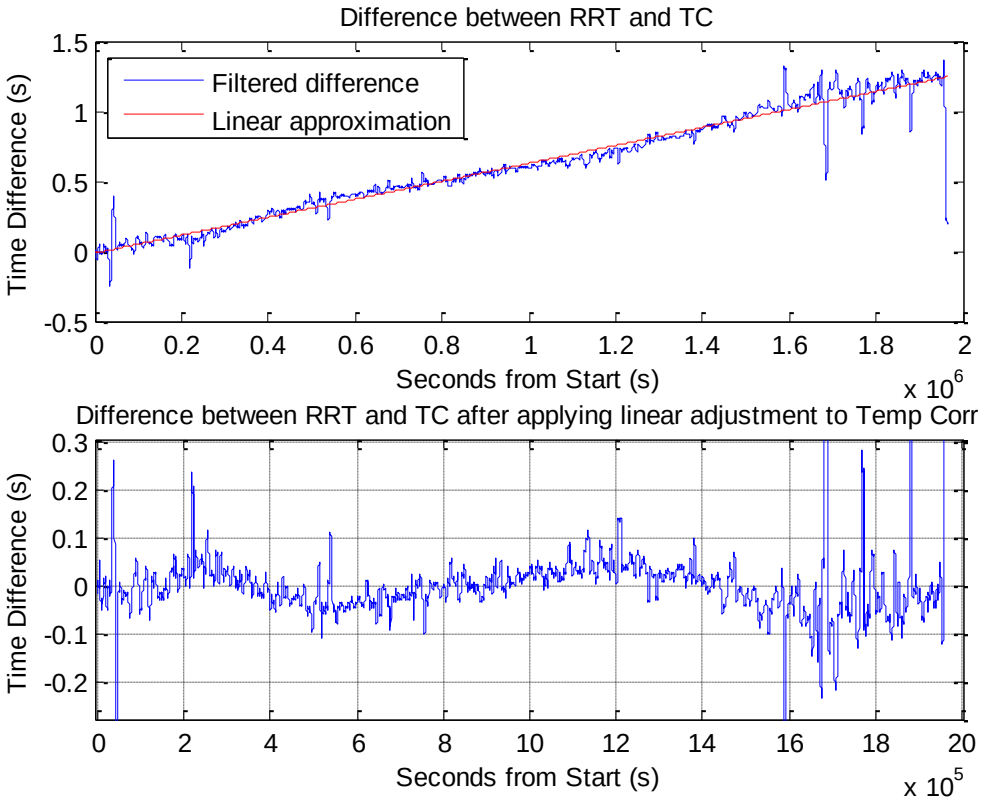


Figure 104. Difference between RRT and Temperature Correction

Approximating the temperature correction error growth using a straight line results in an error of under 100ms. Therefore, adding this linear approximation based on the difference between temperature correction and the RRT is a technique that could be used to improve the time estimation throughout the data. However, this assumes prior knowledge of the RRT results. As the RRT can breakdown due to its inherent noise, this information may not be available. An effort was therefore made to filter the RRT within the main processing algorithm in order to prevent the technique failing.

The filtering would take advantage of the short-term accuracy of the temperature correction but also maintain accuracy in the long-term by using a long averaging window only on points that are within a certain range. Using simple low-pass filtering (averaging) of the raw RRT measurements gave poor results due to the measurements containing many "spikes". Although use of the overall drift could be used by making use of the overall drift observed by the RRT points, the act of adding an adjustment based on RRT to all point meant that the inherently noisier (RRT) time adjustment degraded the time estimate and a more intelligent form of filtering was required.

7.6.3 Filtering using Points of High Confidence

In an effort to benefit from both the temperature correction technique low-noise as well as the RRT's ability to counter long-term drift, it was decided that the best way was to find RRT solutions that had neighbouring points within certain bounds of each other such that there was a very small risk that they were all in error by exactly the same amount. For example, if the RRT solution indicates a time error of 50ms and the next 2 solutions are also 50ms there would be a high confidence in that figure.

The RTC with temperature calibration alone is far more accurate, in the short term, than the noise inherent in the RRT measurement. A conservative value of 2ppm is used for the drift of the calibrated RTC and using time offsets of 10ms, the RRT resolution is 33ppm while its accuracy can actually be much less. The RRT corrections are therefore treated as supplemental adjustments made to the temperature correction, rather than taking over. So, only occasional updates of the measured time drift are required from the RRT and the constraints on what constitutes a "healthy" RRT estimate can be set conservatively - quality over quantity.

The resulting algorithm looks for points in the measurement where the difference between neighbouring measurements agree (3 estimates in a row) is less than or equal to the resolution of the RRT measurement (nominally 10ms). The filtering effect is seen in Figure 105 as a filtered estimate follows the general drift measured by the algorithm.

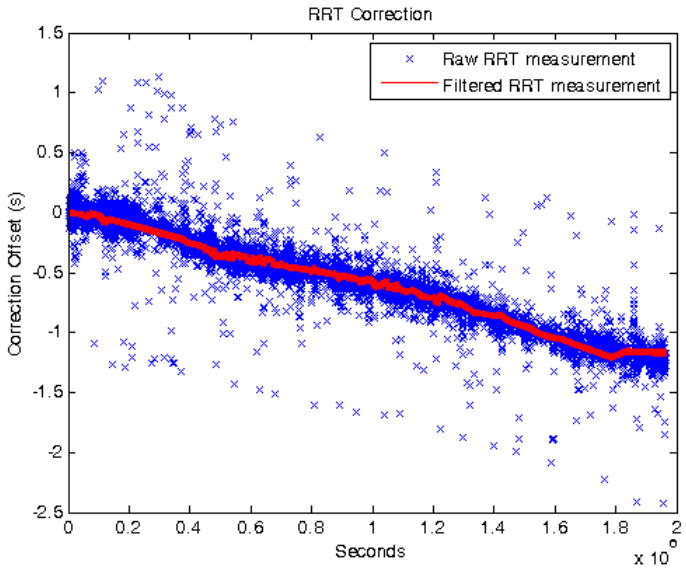


Figure 105. Raw and Filtered RRT using Points of High Confidence

The overall estimate for time correction uses the filtered RRT points to linearly adjust all time offset estimates up to that point in order to give the RRT value. This is very similar to how the end timestamp RTC error is applied to the entire dataset only the RRT can apply the same correction to many points throughout the data. The net result is that the temperature correction points have their long-term drift corrected without losing their short-term accuracy. This is seen, using the same dataset, in Figure 106, where not only the overall drift/error of 1.3s is applied, but numerous points throughout the dataset have been adjusted wherever there were RRT point of high confidence. This means the time correction has been adjusted for long-term drift and is accurate throughout and not just at the beginning and end of the dataset where there were GPS timestamps available to make the drift adjustment.

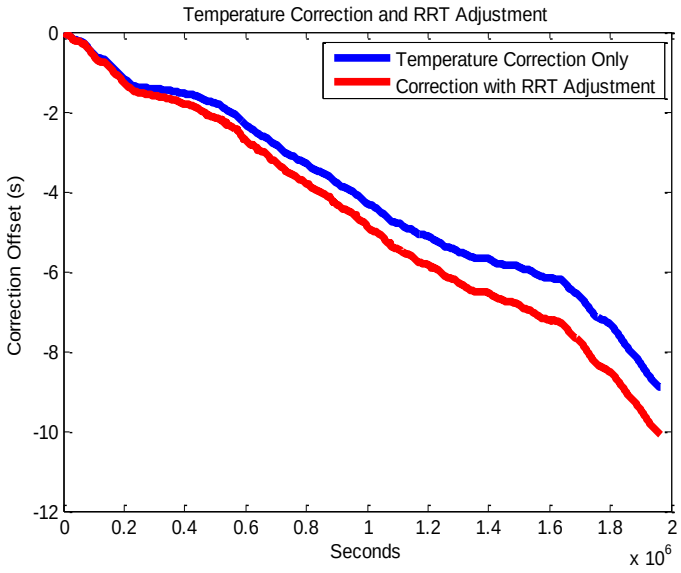


Figure 106. Temperature Correction with/without Filtered RRT Adjustment

7.6.4 Results from using RRT Adjustment

The movement in East direction for the given dataset is shown without RRT adjustment applied (Figure 107) and with the adjustment (Figure 108). A point to note about this dataset is that the end timestamp was not recorded and so the time estimate is not corrected as usual in the TrackTag system. For that reason, the time estimate will get gradually worse as the processing moves from start to finish. (This is what made this a useful dataset for this particular research).

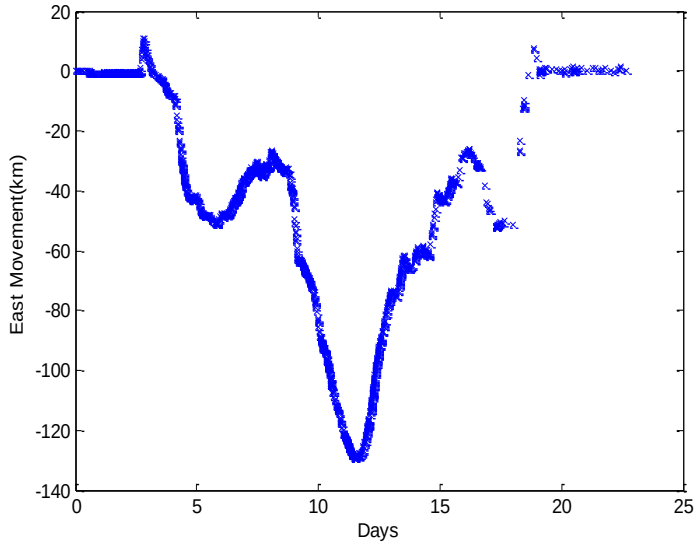


Figure 107. East Movement without RRT Adjustment

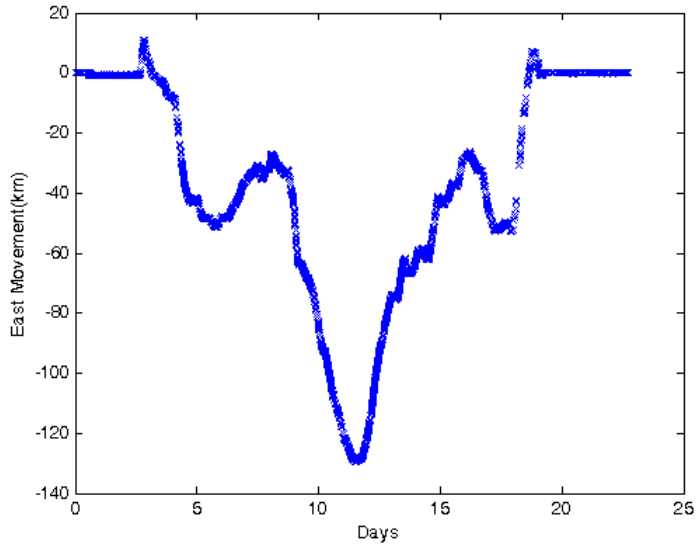


Figure 108. East Movement with RRT Adjustment

Although the traces look similar, it can be observed that the performance nearing the end of the data does get noticeably better with the additional RRT adjustment. For a

clearer picture of this, the two plots are repeated but zoomed in to the period near the end of the dataset, where the time error will have been greatest according to what was observed when looking at the RRT adjustment. The zoomed traces shown in Figure 109 and Figure 110 highlight the difference the RRT adjustment made to the position estimates. There are far more successful navigation solutions and the solutions also have greater accuracy.

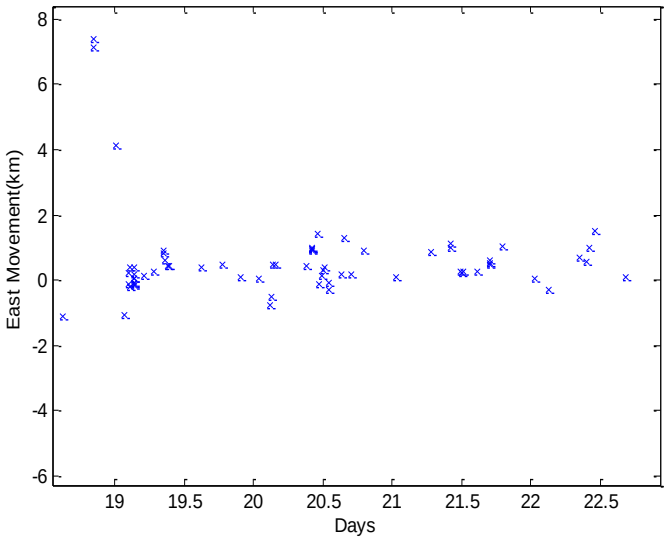


Figure 109. East Movement without RRT Adjustment (zoomed in to end of dataset)

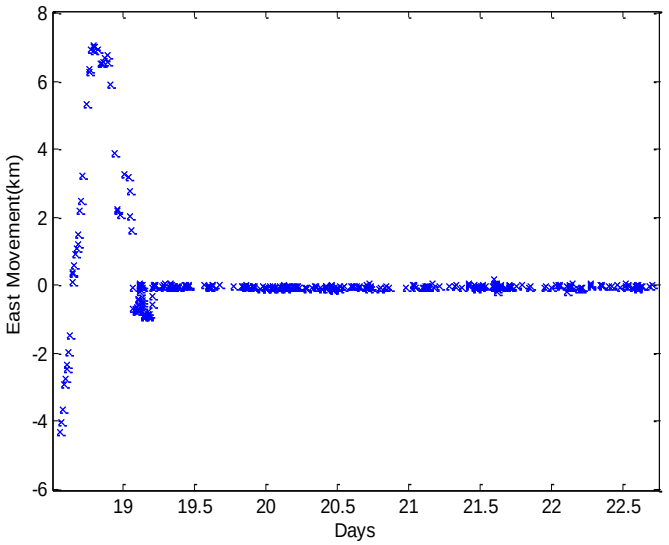


Figure 110. East Movement with RRT Adjustment (zoomed in to end of dataset)

7.7 WAAS Signal Pattern matching

Although there was a solution to the long-term RTC clock drift for TrackTag based on what is described as the Residual Reduction Technique (RRT) discussed in the previous section, there was a problem in one particular application area. Use of TrackTag in the Amazon Rainforest not only presented challenges with respect to acquiring weak signals but also presented a problem when using the RRT algorithm.

The RRT algorithm relies on over-determined solutions to estimate the time error for the particular snapshot in question. So, if most snapshots struggle to acquire enough observable signals to allow this, as they do under the rainforest canopy, the RRT can no longer be relied upon.

When looking for another option to enable real-time time offset estimation, the focus of research turned to WAAS satellite signals. Because they had already been used to augment signal detection (by adding an additional signal) it was already known that the signals could be detected by the TrackTag system. The detection to enable time estimation had to take this further, however, by proceeding to demodulate to WAAS data-stream – something that had never been necessary for GPS signals in the TrackTag system.

7.7.1 Hypothesis

With snapshot lengths of around 26ms data bit matching against GPS signals is not practical due to the bits being 20ms long (50Hz). WAAS signals, on the other-hand, are transmitted at 500 symbols per second. This offers the potential for around 12 symbols to be matched.

As the WAAS broadcast data bits are available in files downloadable over the internet, the idea was to find the location of the snapshots received WAAS bits in the archive file and hence calculate when they were transmitted.

7.7.2 Statistical Analysis of WAAS symbol patterns

The statistical spread of the potential bit patterns found in the broadcast WAAS signal is seen in Figure 111. This was generated using sample "Daily Broadcast WAAS" files downloaded as they would be in the system. The distribution shows the frequency of

every possible potential 12-bit pattern in terms of percentage of total patterns possible in one day. It indicates that around 24% of 12-bit patterns transmitted by the WAAS satellite were all ZERO (0x000). 3.6% of patterns were all ONE (0xFFF).

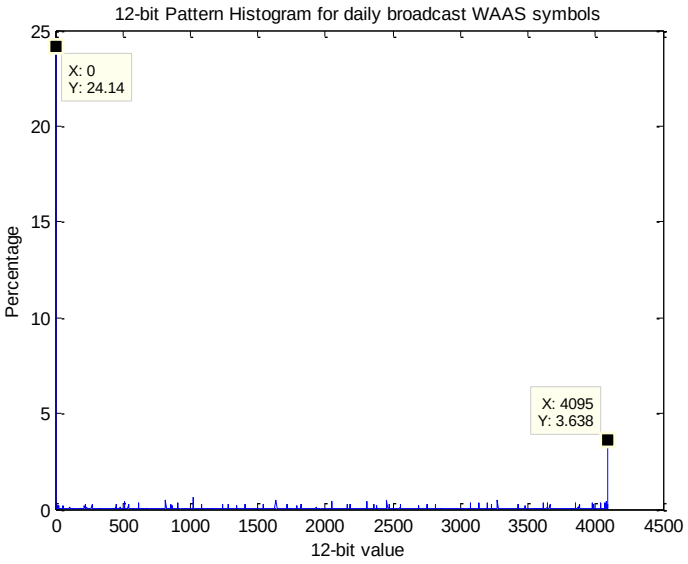


Figure 111. 12-bit Pattern Histogram for daily broadcast WAAS symbols

Due to the TrackTag snapshots being so short, we will be unable to synchronize to the incoming data stream and therefore will not be able to resolve the BPSK phase ambiguity. The 12-bit pattern will therefore effectively be 11-bit as we must consider that the assumed polarity could be, in fact, 180° inverted. This means that a pattern and the corresponding pattern with all bits inverted must be considered as the same.

The effective pattern histogram therefore combines the values for all ZERO (0x000) and all ONE (0xFFF) and all values in between. Figure 112 shows the histogram of non-ambiguous matches for the receiver without knowledge of bit polarity. As expected, the percentage of ALL ZERO (or ALL ONE) patterns is equal to the addition of both corresponding values in the ambiguous case, i.e. 27.8%.

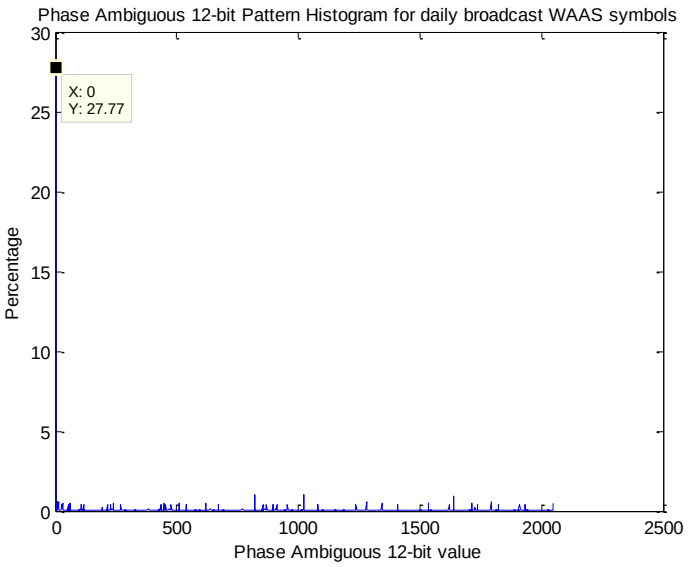


Figure 112. Non-Phase-Ambiguous 12-bit Pattern Histogram for daily broadcast WAAS symbols

This, in summary, suggests that there is enough spread of possible values to be received as long as the ALL ONE (or ALL ZERO) case is ignored. The pattern with the next highest frequency of occurrences is present less than 1% of the time. So, perhaps nearly 1 in 3 snapshots would not be useable as WAAS correction points due to the WAAS pattern being all zero (or one). Just as in the RRT algorithm for adjusting time correction estimate, only an occasional adjustment is necessary to correct for the long-term drift of the RTC and so the WAAS pattern matching was deemed to merit further study.

7.7.3 Algorithm Description

While attempting to match WAAS bit patterns the main challenge was in how to handle the fact that there could be many matches thus giving an ambiguous solution. The following steps describe how the algorithm works and how it deals with resolving ambiguities that arise in the solutions.

1. Calculate the potential RTC correction error for all snapshots. This is based on an estimate for potential error in the temperature reading (2°C) and what that equates in time given the tag's temperature calibration curve at the each snapshot temperature. An additional 2ppm drift (very conservative) is added to

allow for long-term drift. The potential RTC correction error for each snapshot determines the size of WAAS pattern matching window to be used.

2. Extract the WAAS signal pattern detected and bit-phase (1 of 2ms for WAAS signals) from the signal detection record.
3. Estimate the WAAS signal Time-of-Flight and therefore Time-of-Transmission based of estimated tag position and WAAS satellite orbital position at assumed time. The last known tag position and can be used here as a first approximation as it is assumed that any tag movement will be near perpendicular to the line-of-sight as well as very small relative to the WAAS altitude and will therefore have negligible effect on the transmission. The WAAS orbital position can based on uncorrected time without significant effect on the estimate transmission time as (illustrated in Figure 85) the satellite's relative movement runs over a slow, approximately daily, cycle and distance spanned is under 100km during the day.
4. Estimate position/time of received WAAS symbols within the timeframe of the day in question.
5. Extract relevant WAAS symbols for seconds of interest in the daily WAAS broadcast data file available on the internet. These seconds have to include the uncertainty of the RTC time correction.
6. Find symbol pattern positions that correlate exactly with the detected WAAS pattern (or its bit inverse as received signal polarity is not known).
7. Filter out any matches that fall outside the potential RTC correction error.
8. Place all potential WAAS correction time offsets into 2-dimensional "WAAS correction matrix".
9. Use the matrix to check ambiguous corrections against unambiguous instances that appear in the correction matrix. Discard any solutions that fall outside the differential error allowed between that point and an unambiguous solution.
10. End

The "WAAS correction matrix" basically acts as a filter that removes any solutions that cannot fall within the clock drift bounds when compared to any other solutions in the matrix. The matrix was set to allow up to 200 ambiguous solutions per snapshot and stored the results for up to 5 neighbouring snapshots that were deemed capable to providing a WAAS time correction.

The matrix is moved through the data while constantly checking validity of each potential solution and removing invalid solution as it went. The end result was that all solutions that were left satisfied the assumed maximum drift possible.

7.7.4 Results from WAAS Time Correction

Figure 113 shows the WAAS correction points that were calculated for the "Romina" dataset which was taken from a study in the Amazon, tracking Tapir. There were no GPS signals detected during the first half of the data and the signal quality got gradually better from the halfway point. This explains why we only see WAAS time correction points in the latter half of the dataset.

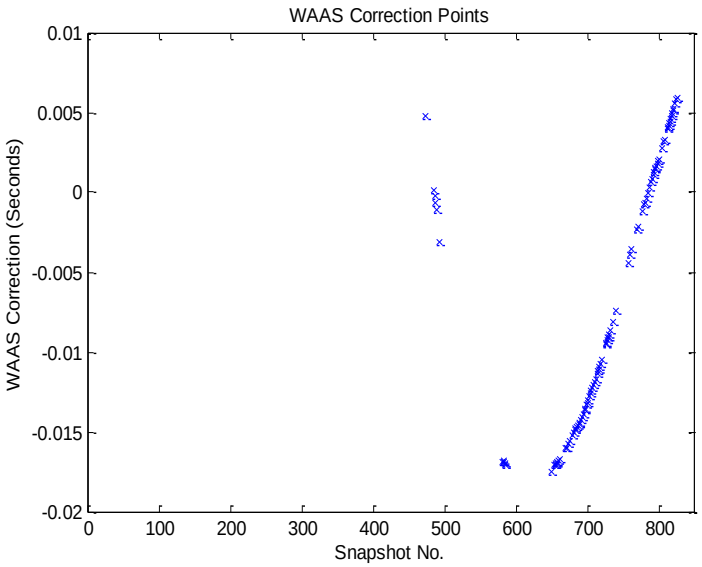


Figure 113. WAAS Correction Points for "Romina" Dataset

If the RTC temperature correction had been perfect, the WAAS adjustments would have all been zero. As it was, even though this is a relatively short dataset (approximately 9 days) the actual WAAS correction varies up to 17ms. It would also be expected that the endpoint approaches zero due to the normal endpoint timestamp adjustment carried out by the TrackTag system.

Because there is what appears to be a fairly linear region between snapshots 650 and 800, Figure 114 is plotted to examine the correction in more detail.

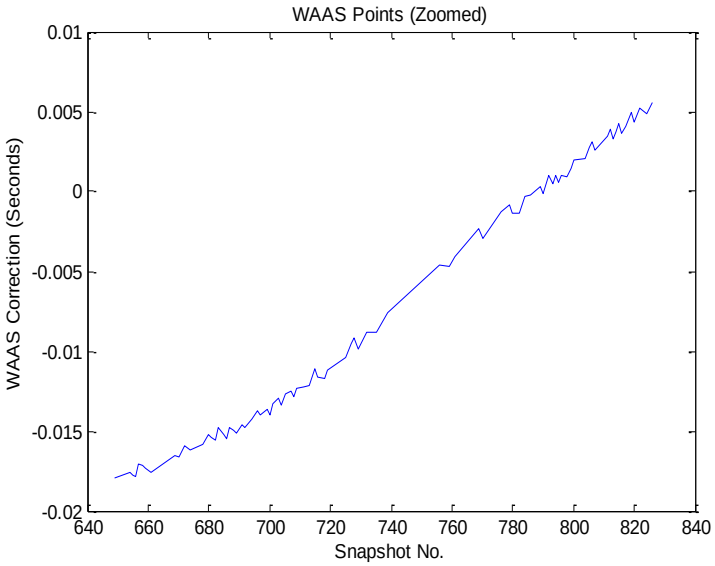


Figure 114. WAAS Correction Points (zoomed)

The variability of the time correction seemed to "zigzag" a little and so the difference between successive corrections was plotted (Figure 115). This showed seemingly random noise that was usually bounded between 0 and 1ms and only exceeded 1ms when there was a relatively long duration of time since the last time correction point. It could be explained by the WAAS, BPSK code-phase not being taken into account in the time estimation. The modulo 1ms, 1023-chip code-phase used in all the navigation calculations was not taken into account when estimating the WAAS time correction as only the WAAS pattern match and bit-phase were used. It was not foreseen that the algorithm would work well enough to enable this level of (sub-millisecond) correction.

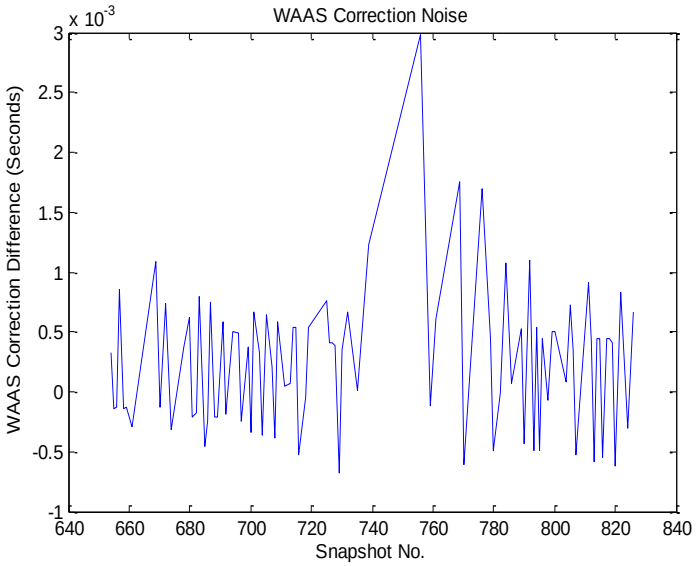


Figure 115. WAAS Correction "Noise"

By making minor modification to the algorithm, the WAAS pattern, bit-phase and code-phase were all included in determining the precise timing of the snapshot. The improvement to the time correction noise can be seen in Figure 116. Most of the noise was reduced substantially, and the noise is more random overall and so can therefore reasonably be assumed to be accounted for by the actual error in the original RTC correction. The large spike is only 3ms and occurs following a fairly long period of time without other WAAS corrections and so also falls within acceptable limits.

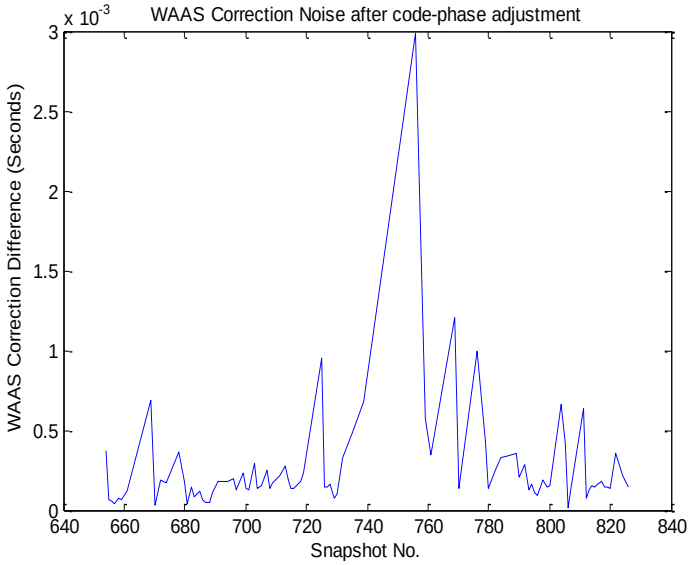


Figure 116. WAAS Correction "Noise" with Code-Phase Adjustment

What this final adjustment achieved can also be observed in Figure 117 which corresponds to the same corrections shown in Figure 114. The resulting curve is noticeably smoother. Observation of how applying the code-phase correction has resulted in noticeably improved performance suggests that the absolute accuracy of the time offset measurement can be related to the sampling frequency. As the sample frequency for TrackTag has been 5MHz throughout this research the time offset measurement accuracy is around 200 nanoseconds.

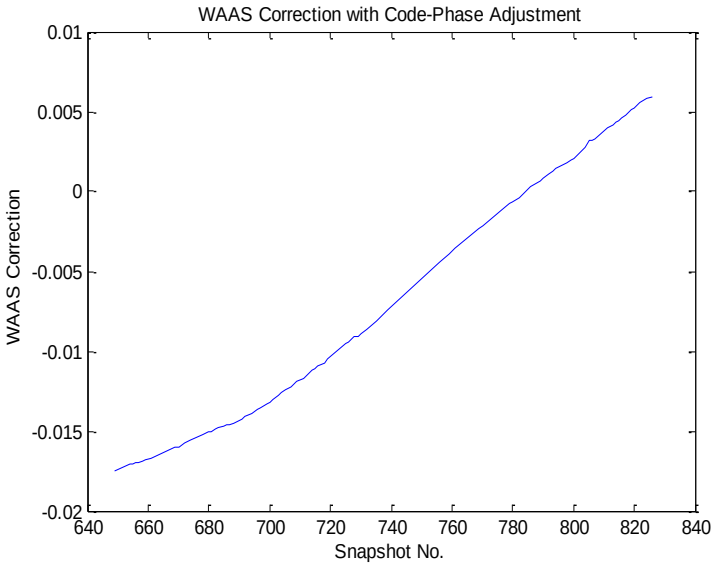


Figure 117. WAAS Correction Points with Code-Phase Adjustment

The WAAS time correction demonstrated synchronisation of time to sub-millisecond accuracy. This will be a very powerful technique for resolving time error on long-duration data. The effect can be seen on a relatively short (9 day) dataset where the number of successful navigation solutions rose from 183 (as observed in Figure 92 in section 6.2.5) to 239 when WAAS time correction was introduced. This represents an improvement in availability from 40% to 55% when you consider that the tag was not actually taken outside until around snapshot 450. The accuracy was also noticeably improved (Figure 118) compared to before.

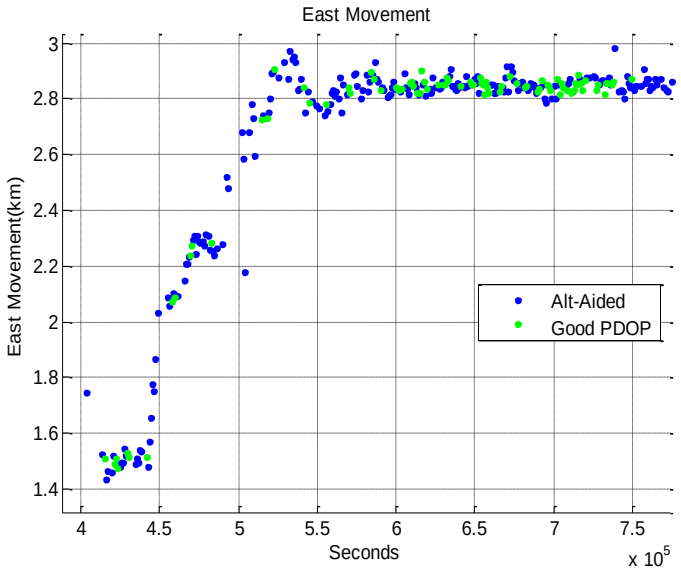


Figure 118. "Romina" Dataset East Movement

Chapter 8: Results & Conclusion

8.1 British Antarctic Survey (BAS) Results

This research started with Albatross data taken by BAS during 2004 in the South Atlantic Ocean. While weak signal detection should not have been a major consideration, these datasets were taken before the antenna integration work had been done so the tag hardware offered poor signal reception as the main gain of the antenna was basically pointing backward from the bird, sometimes even slightly downward!

The duration of the trips varied a lot but did not exceed a month. We therefore got useful results using only temperature correction but did notice a different when implementing what were the early revision RRT solutions.

Figure 119 shows the track for an Albatross with different colours indicating different days. Google Earth file format was used as a convenient way of superimposing track data on to maps. The main island, seen to the right (East) is South Georgia. However, the track starts and ends on a small island off the Northwest tip of South Georgia known as "Bird Island" as this is where a lot of wildlife research is undertaken.

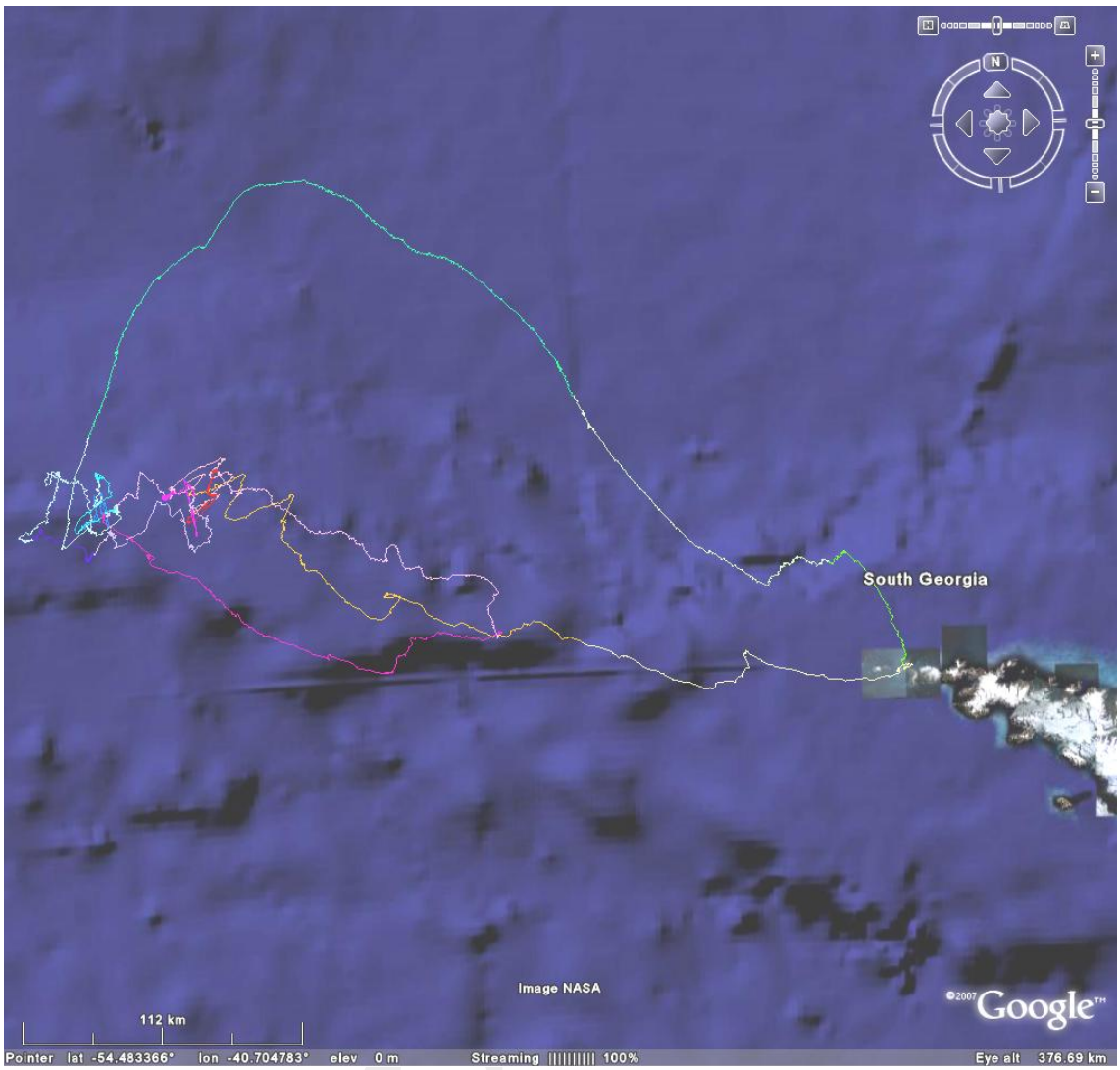


Figure 119. Albatross Track in South Atlantic

The track shown in Figure 120 was made by a King Penguin. A small part of this track is also zoomed-in to show the accuracy achieved. By the time this data was being processed the RRT technique was more mature and was used routinely. Time correction was not the major challenge with penguins though – it was the additional requirement for inhibiting snapshots while the penguin was underwater. Although not part of this thesis, the algorithms had to cater for the fact that a recently developed "salt-water switch" could make the snapshot cadence effectively random.

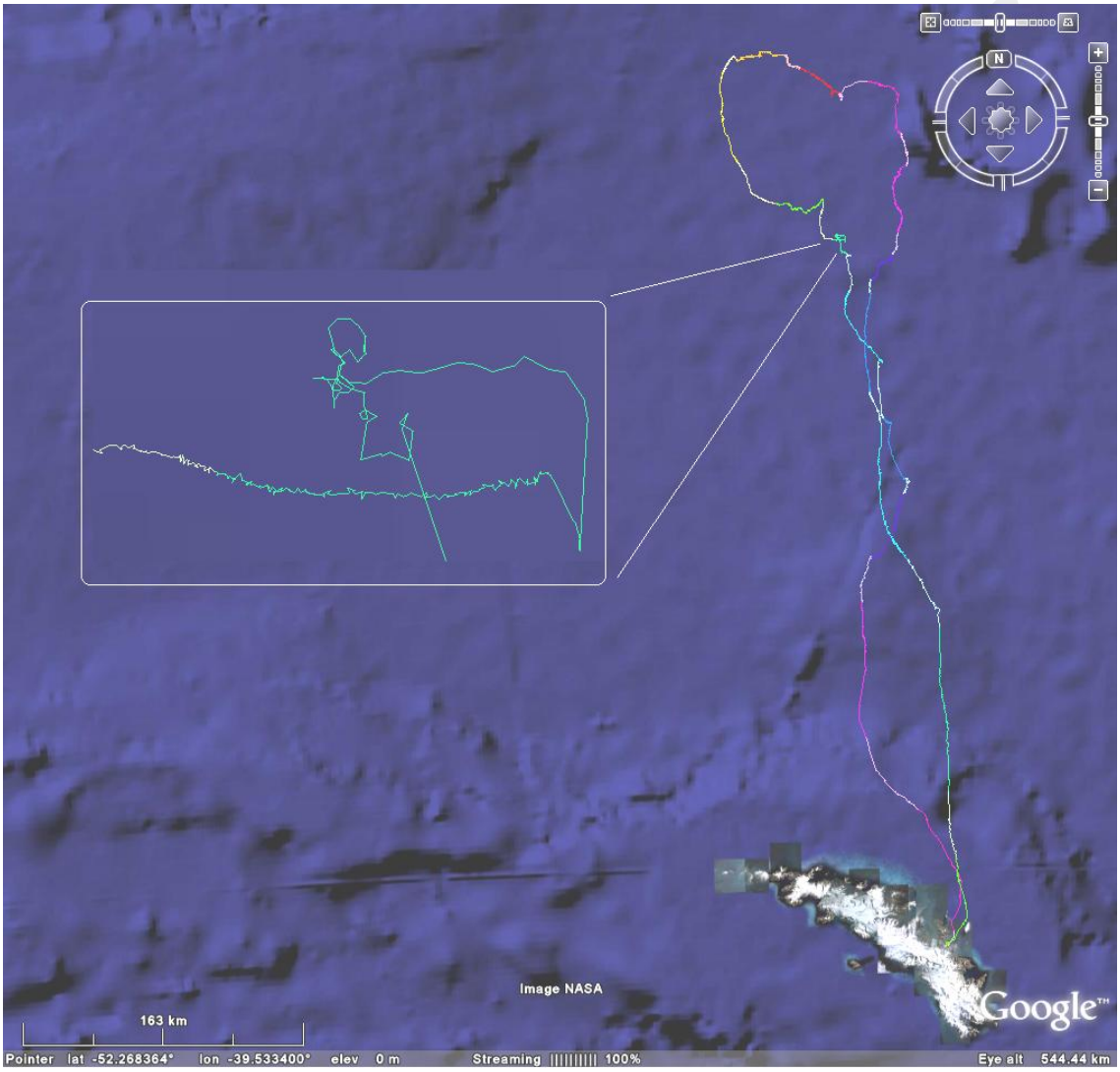


Figure 120. King Penguin Track in South Atlantic

8.2 University of Bangor Results

Leatherback Turtles were studied using TrackTag in 2006. There was particular interest In where the turtles would come ashore as the researchers were studying the effects of human behaviour on the life of these creatures around the popular holiday destination of Zante. Figure 121 shows one example of the tracks turtles made which clearly shows the location the creature came ashore.

Figure 121. Leatherhead Turtle Track in Zante

This research was well received and resulted in publication in The Journal of Experimental Marine Biology and Ecology [72].

Incidentally, if you look closely, you can also see the track venture a number of kilometres inland around the airport and into the main town. This was in fact due to the researchers not bothering to download, and terminate the deployment, until they got back to the office!

8.3 Moore Foundation / University of Austin Results

The work done in the Amazon Rainforest was by far the most challenging. Weak signal detection was vital in order to detect the signals through dense forest areas. Also, the deployments were up to 1 year duration. Due to the lack of signals, the RRT technique did not perform well and this is what necessitated the research and development of WAAS pattern time correction which resulted in significant improvement to the accuracy of the tracks. An availability of position estimates with 2dRMS accuracy within 50m of >50% is observed over different datasets relating to various species.

Although, at the time of writing, there have been no publications on the work undertaken by the biologists there using TrackTag, a number of papers are expected soon for work done tracking Tapir and Jaguar. Figure 122 shows one example track for a Jaguar tagged in the Peruvian Rainforest near Machu Picchu.

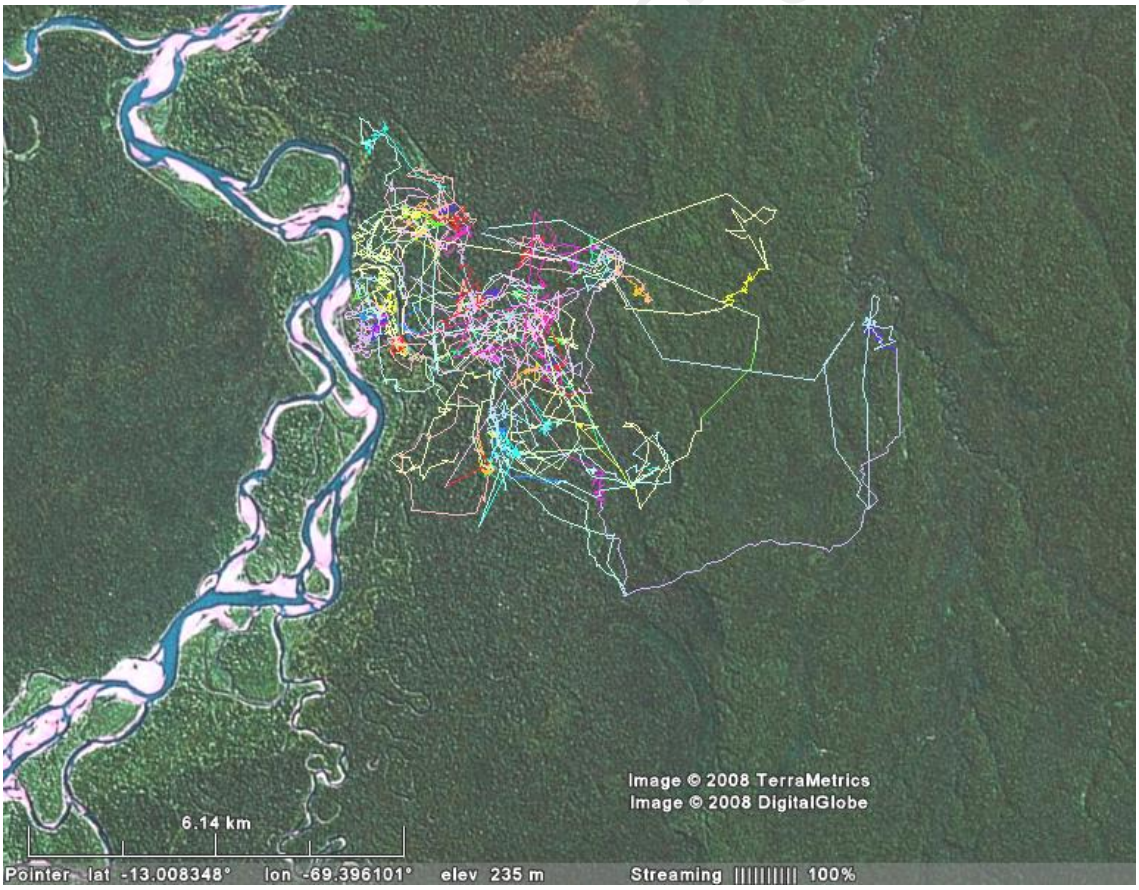


Figure 122. Jaguar Track in Amazon Rainforest

8.4 Ongoing Research using TrackTag

At the time of writing, TrackTag is currently being deployed on more species in the Amazon as well as Geese flying over the Himalayas. Further development involving shrinkage to even smaller dimension means the system will be capable of providing a method for researchers around the world to acquire positional data that has not been technically feasible prior to the TrackTag's development.

8.5 Conclusion

Research and development of the TrackTag system was a complex task not only due to individual parts of the system but also in the way they all interact. Throughout the research the problems encountered often depended on improvements in other areas which meant the research did not follow a well defined route. The thesis has been written in a manner that attempts to show a sequential flow to the work but in reality this was far from the truth!

The process of writing-up the work, however, has clarified much of the work and I believe presents a coherent flow from general background through to results. The work was, of course, also carried out in collaboration with a commercial entity, NAVSYS, without whom none of it would be possible, if for no other reason than they hold the a patent that underpins the entire concept on which TrackTag is based.

The WAAS correction, in particular, is probably unique and could offer more benefits moving forward as, at the time this research was done, the WAAS constellation was still in its infancy and so WAAS coverage should improve over the next few years.

Overall, I believe the work detailed in this thesis has already benefited the animal tracking community as it has already enabled biologists to do some work that had been unfeasible due to limitations of other tracking technologies. There should be more wildlife studies based on TrackTag making it to print in their respective journals as NAVSYS have sold more systems into that market.

Appendix A: Position Estimation

The position estimation algorithm used in this research is a relatively basic design. There are many enhancements that could be made to improve the algorithm to make it more robust against rogue measurements, e.g. weighting of the measured ranges based on signal strength and various methods for filtering position estimates (usually involving the use of Kalman Filters in GPS).

Using a simple algorithm is necessary for clear analysis of the performance of other parts of the system and it was therefore considered best practice to keep the position estimation algorithm as simple as possible for use with the signal detection and time correction research. The method for calculating position was determined at the start of the research and was kept unchanged throughout. It is based on the theory described in this appendix.

A.1 Pseudo-range

The pseudo-range is the apparent range to an individual satellite. This is made up from the actual range plus some error terms as described in equation (28) [17]

$$\rho^{(k)}(t) = r^{(k)}(t, t-\tau) + c\left[\delta t_u(t) - \delta t^{(k)}(t-\tau)\right] + I^{(k)}(t) + T^{(k)}(t) + \varepsilon_\rho{}^{(k)}(t) \quad \textbf{(28)}$$

The equation defines the measured pseudo-range to satellite ‘k’, at time epoch ‘t’. The actual range is given as ‘r’ which is a function of current time epoch and the transmission time, ‘τ’, subtracted from the current time. The error terms are defined as follows;

$\delta t_u(t)$: Receiver clock bias at current time epoch.
$\delta t^{(k)}(t-\tau)$: Clock bias for satellite ‘k’ at time of transmission.
$I^{(k)}(t)$: Ionospheric propagation delay for satellite ‘k’ at current time epoch.
$T^{(k)}(t)$: Tropospheric propagation delay for satellite ‘k’ at current time epoch.
$E_\rho{}^{(k)}(t)$: Modeling errors (such as orbit prediction error) and un-modelled effects (such as receiver noise)

Assuming we can correct for the satellite clock drift and the atmospheric errors. We are left with the following simpler equation;

$$\rho^{(k)}(t) = r^{(k)}(t, t-\tau) + c\delta t_u + \varepsilon_\rho{}^{(k)}(t)$$

(29)

Equation (29) shows that the pseudo-range, ρ, will differ from the actual range, r, only by two terms.

1. A distance related to the receiver's clock error.
2. The residual errors mainly due to inaccuracy of signal detection.

A.2 The Geometry Matrix

The creation of the geometry matrix, is described in equation (30)[17]. It is created using the LOS (line-of-sight) unity vectors from the assumed receiver position and each satellite (denoted 1,2… n in this representation).

$$G = \begin{bmatrix} -\cos(Az_1).\cos(El_1) & -\sin(Az_1).\cos(El_1) & \sin(El_1) \\ -\cos(Az_2).\cos(El_2) & -\sin(Az_2).\cos(El_2) & \sin(El_2) \\ -\cos(Az_3).\cos(El_3) & -\sin(Az_3).\cos(El_3) & \sin(El_3) \\ \cdots & \cdots & \cdots \\ -\cos(Az_n).\cos(El_n) & -\sin(Az_n).\cos(El_n) & \sin(El_n) \end{bmatrix}$$

(30)

The pseudo-ranges are compared against the predicted actual range from the assumed position to the satellite's orbital position. The difference between the two values is considered to be the "residual error".

The vector of residual errors is then fed into an iterative process which, along with the geometry matrix (for direction), aims to find the geometric position that minimises the residual errors and is therefore proposed as being the navigation solution.

A.3 Dilution of Precision

There are various forms of Dilution of Precision (DOP) that can be useful in GPS. The one of interest to this research is the Position DOP, or PDOP. It can be calculated directly from the geometry matrix by taking the diagonal elements of the "H-matrix"[48] which is calculated as shown in equation (31).

$$\begin{aligned} H &\equiv \left(G^T G\right)^{-1} \\ PDOP &\equiv \sqrt{H_{11} + H_{22} + H_{33}} \\ HDOP &\equiv \sqrt{H_{11} + H_{22}} \\ VDOP &\equiv \sqrt{H_{33}} \end{aligned} \tag{31}$$

The PDOP value relates to all 3 physical dimensions as it is a measure of the precision in 3D coordinates. The HDOP and VDOP values refer to horizontal and vertical precision respectively.

Appendix B: Platform Development

There were major design challenges that had to be tackled before the bulk of the research could take place. This involved the development of the TrackTag product.

The product development was considered as being the research platform development. The development was also required to meet the commercial goals for the company during the first 2 years. This, in turn, provided valuable field data that was used in the research. The latter years of the research would see the company rely more and more on the results from the research & development undertaken.

This appendix aims to outline what challenges were met while creating the platform to be used for the development of TrackTag. There were 3 main areas of development and although the algorithm development was considered as being the primary research area, development in that area had to be done early-on simply to get the system up-and-running. There was not a simple sequential and linear process.

- Hardware development
- Software development
- Algorithm development

The algorithm development involved is discussed in depth within the research chapters. The hardware and software development is described in this appendix as the "Platform Development". The main reason for this is that the platform development is required for data acquisition. The research required real-world data in order to make decisions on what capabilities would benefit the customers.

Although the platform development was not considered highly innovative in itself, there was some innovative work done. The platform accounts for a large part of the research time and its operation is fundamental to the design of the algorithms.

The TrackTag system consists of the tag(s) and an Interrogator unit. The Interrogator unit is required to interface between the tag and the user's PC. Figure 123 illustrates the basic setup. In addition to the data interfacing between PC and tag, the Interrogator also

provides a battery charger (used only when a rechargeable battery is connected to tag), and a conventional GPS receiver – hence the need for an antenna.

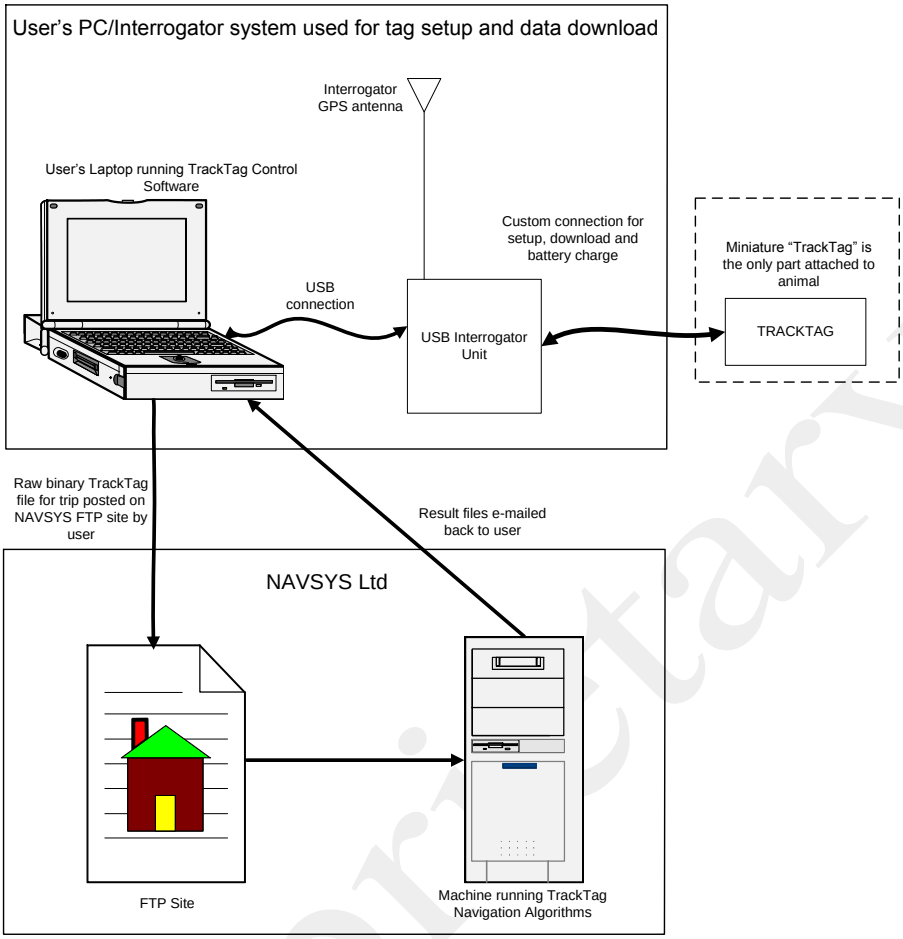


Figure 123. TrackTag System

The overall system works in 3 phases;

1. Initialisation
2. Trip
3. Download

It's only during the "Initialisation" and "Download" phases that the Interrogator is required as the "Trip" phase refers to the time the tag is out recording data in the field.

During Initialisation, a PC is used to provide a user-interface to the Interrogator. The Interrogator, in turn, utilizes an internal OEM GPS receiver in order to set the clock on the tag immediately prior to being sent out on a trip. It also configures the tag's user-

defined parameters such as the duty-cycle between fixes and has a built-in battery charger that will charge the tag battery if required. All of this has to be controlled by software in the PC.

During a Download phase, the PC is used to control the time-stamping (this time to mark the end of the trip) and the download of all recorded data on the tag. The PC generates a raw data file with a header containing a few key pieces of data such as timing and position information for the start and end of the trip. The file is then sent to NAVSYS Ltd where it is loading into a processing PC that will run the algorithms developed during this research. The result, if successful, is a full track of where the tag had gone which is then sent back to the customer in an agreement format.

B.1 Interrogator Development

The Interrogator board was produced from schematic capture through to layout as part of the platform development and has been sold or loaned out to customers located around the world. Because of the strategy of using customer data in development of the algorithm development, it was important to complete the design of the Interrogator early-on. This therefore had to be completed at a very early stage of the research.

B.1.1 Software Development

Just as the Interrogator was required from the start of the research, so was the software needed to control it.

Visual C++ was chosen as the design environment. The reasons behind this were the author's previous experience of C programming and the fact that C is prevalent within the NAVSYS as a whole. The "Visual C" element was considered useful in helping the rapid development of a user-friendly GUI (Graphical User Interface).

In accordance with good software design, the high level coding required for the GUI was kept separate from any low-level hardware control code. The way this was split is detailed in Figure 124. At the GUI level, the user windows (dialog boxes) are partitioned according to function. The code follows that partition and therefore results in a number of files for the GUI code – one per dialog box – and a single file which encompasses the low-level code which calls the driver functions.

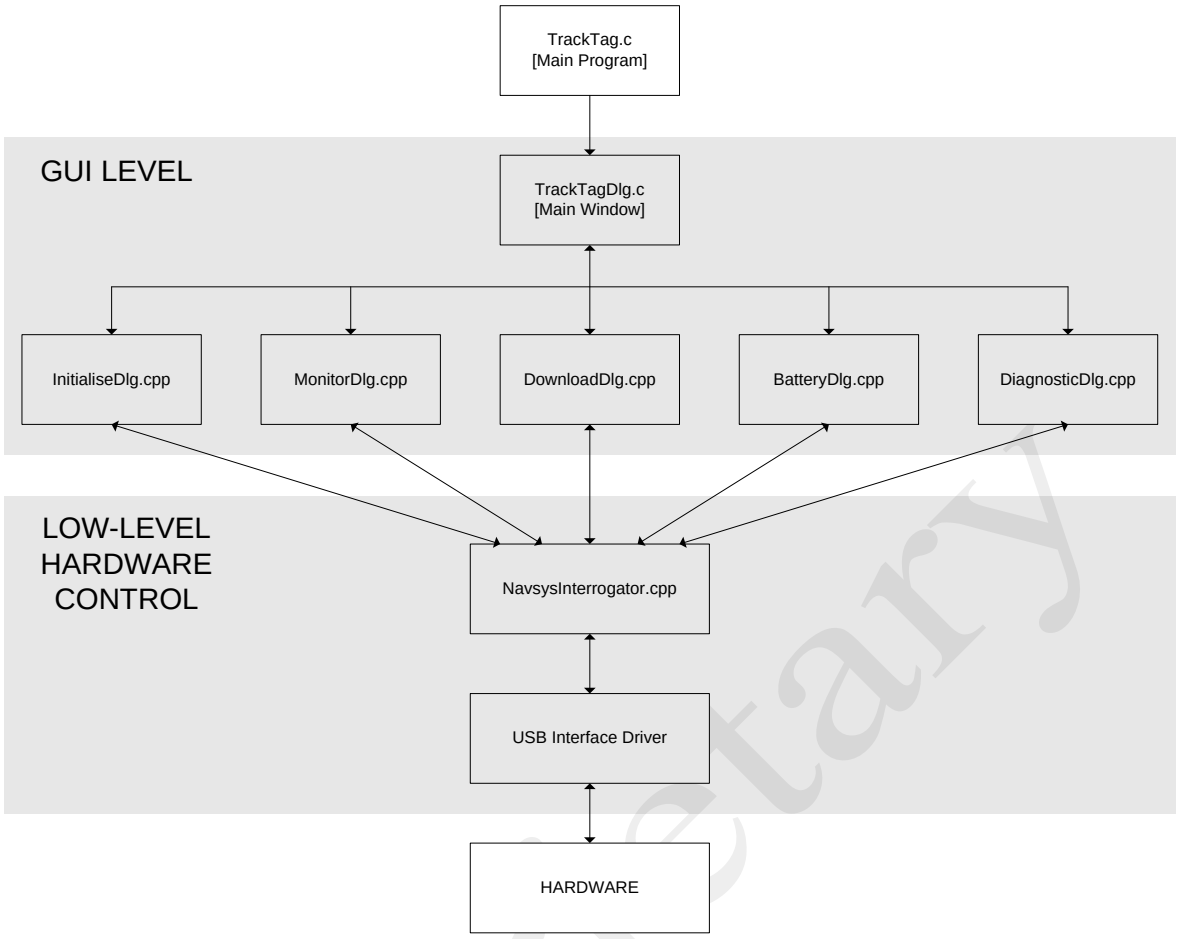


Figure 124. Software Top-Level Design

The Dialog window files correspond with 6 different dialog boxes that can appear, although many other smaller windows can appear but are relatively simple and contain limited control functionality. The appearance and functionality of each of these screens are now explained.

Main Window

Figure 125 shows the main screen that is displayed from launch of the executable. It provides the user easy access to the main elements of the software used to control the TrackTag system.

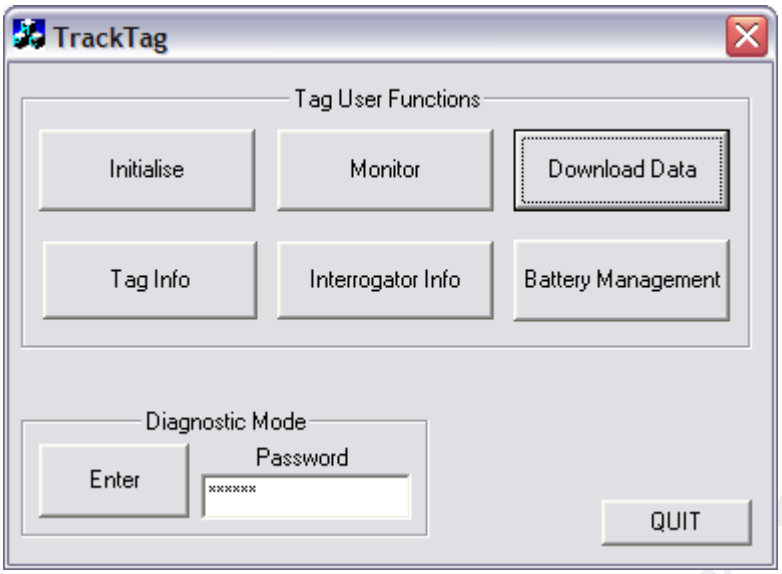


Figure 125. Main Window

There are 6 user functions and the "diagnostic mode" that allows low-level control functions for debug. The 6 user functions are;

1. Initialise: Allows user to erase/check the tag flash and program/check the tag's trip parameters such as wake up rate and snapshot size.
2. Monitor: Displays a scrolling screen which reports a time stamp and snapshot number every time a snapshot is taken (while a tag is connected). This allows the users to see that the tag is taking snapshots at the expected rate.
3. Download Data: Allows user to download the tag data.
4. Tag Info: Reads and displays the tag configuration parameters.
5. Interrogator Info: Reads and displays the Interrogator parameters.
6. Battery Management: When rechargeable battery is installed on tag this displays the charger current and voltage, charge state and can create a log file of these parameters over time.

Initialise

Figure 126 shows the "Initialise" user function window. When this screen is first opened it automatically populates the parameters with those stored in a user-editable configuration text file. This action could be repeated at any time by clicking the "Read Config File" button and ensuring the correct filename is shown in the relevant box.

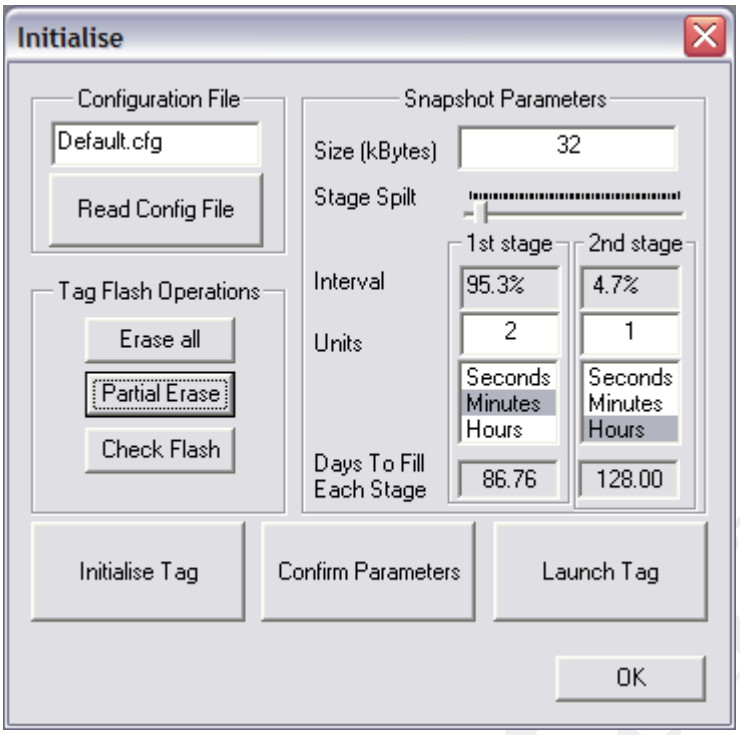


Figure 126. Initialise Window

Within the "Tag Flash Operations" box there were 3 functions;

1. Erase All: Wiped the entire Flash memory on the tag.
2. Partial Erase: Used another low-level function that estimates the amount of data on the Tag Flash by implementing a binary search of Flash blocks checking whether they are clear. This offered a very fast erase if there is only a small amount of data on the Flash.
3. Check Flash: Assumed the Flash was empty. Ran through the entire Flash checking each block was empty.

The "Snapshot Parameters" box held the snapshot configuration variables to be programmed on to the tag. The "Stage Split" enabled the user to program 2 sets of these parameters. The percentage set determined how much of the tag memory was filled before it switched over to the 2nd set of parameters. This was useful to some biologists as they could not be sure when an animal will return and this therefore provided secondary area of the Flash that would be filled at a far slower rate. It used a relatively small amount of the tag memory.

The large "Initialise Tag" button used the displayed snapshot parameters and programmed the relevant parts of the tag with those settings.

"Confirm Parameters" read and displayed the parameters currently on the tag.

"Launch Tag" sent the tag into its trip mode. This effectively sent the tag to sleep, where after it would only wake when the on-board clock had been programmed to wake and store a snapshot of the RF spectrum to Flash. The only way to regain control of the tag by the Interrogator was by requesting to read the tag parameters. This was usually done automatically by opening the "Download Data" via the main screen.

Monitor

The Monitor window (Figure 127) could be used to watch the connected tag's snapshot intervals. The tag sent the snapshot count to the Interrogator (if connected) every time it woke. This screen showed the snapshot number and the time it was awakened. This was very useful as a sanity check for the biologist to see that the tag was running before attaching to an animal.

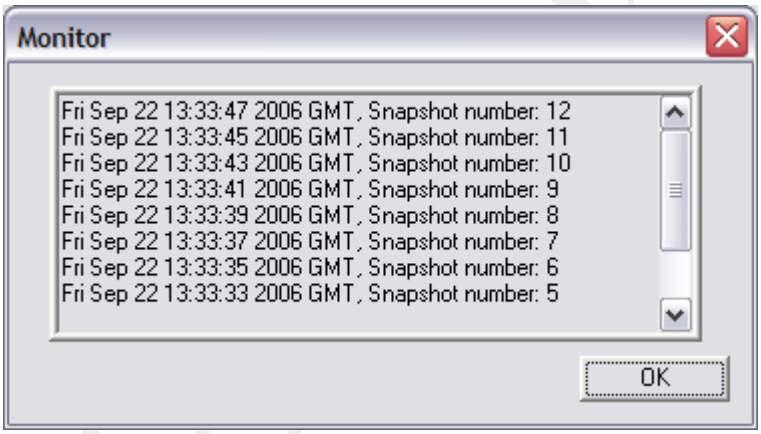


Figure 127. Monitor Window

Download

The download window (Figure 128) reported an estimate of the number of snapshots recorded on the tag;

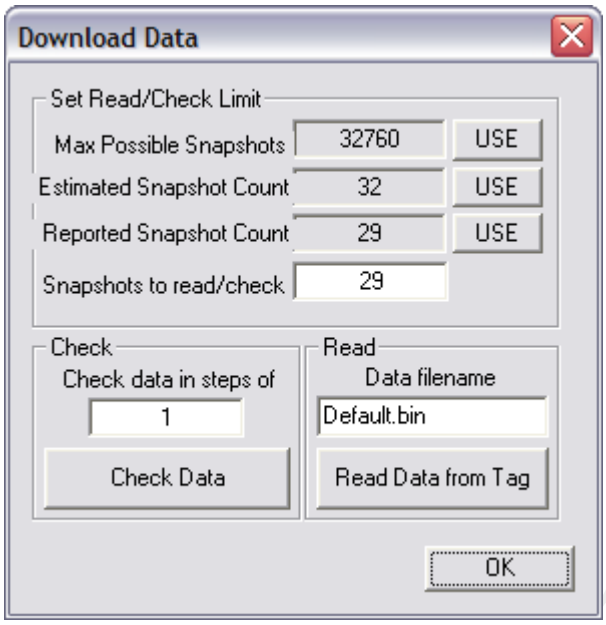


Figure 128. Download Window

"Max Possible Snapshots" was the maximum possible number of snapshots stored on the Flash.

1. "Estimated Snapshot Count" was calculated by running through the Flash memory and detecting the point where the data became ALL ZERO.
2. "Reported Snapshot Count" was what the tag snapshot counter value reports.

Under normal circumstances the 3rd value would be used. However, the user may have had difficulty in retrieving the tag before its battery went flat, in which case the reported snapshot count will have been lost due to the value being stored in volatile memory.

When the reported value cannot be relied upon, the user could use the estimated count. This was calculated by running through the Flash quickly while checking to see if the memory is blank or not and thus estimating where the data ends. The user also had the option of using the maximum possible count (read the entire Flash memory) or they could manually set the number of snapshots to read.

By selecting the "Check Data" button a comma-delimited text file was created that gave each snapshot's time and number. In order to speed up this check, a step size between snapshots could be used. For example, setting a step size of '10' meant that snapshots would be checked in the sequence 1, 11, 21, 31…

Interrogator Info

Interrogator Info (Figure 129) was an information only screen providing various hardware and software serial numbers and version numbers.

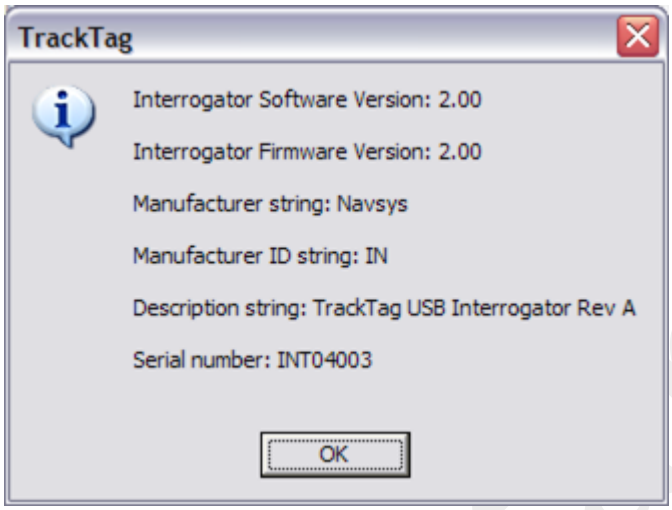


Figure 129. Interrogator Info Screen

Tag Info

Tag Info (Figure 130) was an information only screen.

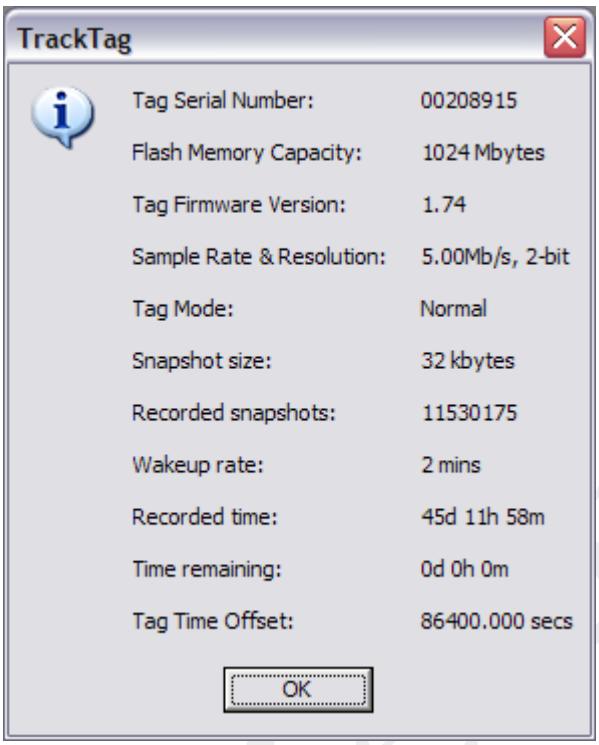


Figure 130. Tag Info Screen

Tag Serial Number:	Unique identification for each tag.
Flash Memory Capacity:	Size of memory used to store data.
Tag Firmware Version:	Version of control logic loaded into tag.
Tag Sample Rate:	Sample rate at which tag would record GPS signals.
Tag Resolution:	Bit resolution to be used to record GPS signals.
Tag Mode:	'Normal' or 'Calibration' mode.
Wakeup Rate:	Duty-cycle at which the tag was programmed to wake.
Snapshot size:	Number of Kbytes that would be stored for each snapshot.
Recorded snapshots:	How many snapshots were currently held on the tag.
Recorded time:	Time period the recorded snapshots covered.
Time until memory full:	How long the tag could go before its memory runs out.
Tag Time Offset:	Current difference between tag RTC and GPS clock in Interrogator.

Battery Management

The battery charge cycle started by applying a constant charge current which resulted in a gradual increase in the charge voltage to around 4.2V while the current was roughly equal to the charge current specified in the configuration file. Just prior to this final voltage being reached, the charger circuit went into "constant voltage mode" and started to reduce the charge current. When the current then fell below 10% of the specified value, the cycle was considered complete and the indication of charge state would read "FULLY CHARGED".

The battery's voltage and charge current (in mA) were displayed in the battery management window (Figure 131). A log file could be created using the "Start logging" and "Stop logging" buttons. This created a text file which can be used for battery charge diagnostic purposes.

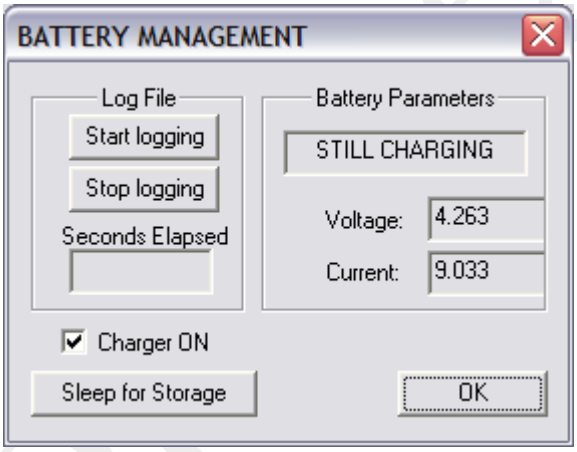


Figure 131. Battery Management Window

This window also allowed the user to switch the charger circuit ON/OFF and send the tag to sleep (although that is usually done as the application software is closed anyway).

B.1.2 Low-Level Control

A flow chart showing the usual function sequence is given in Figure 132. Most of the complexity involved in setting up the tag is hidden from the user. This is, of course, the main reason for creating the GUI environment for who are likely to be non-technical customers.

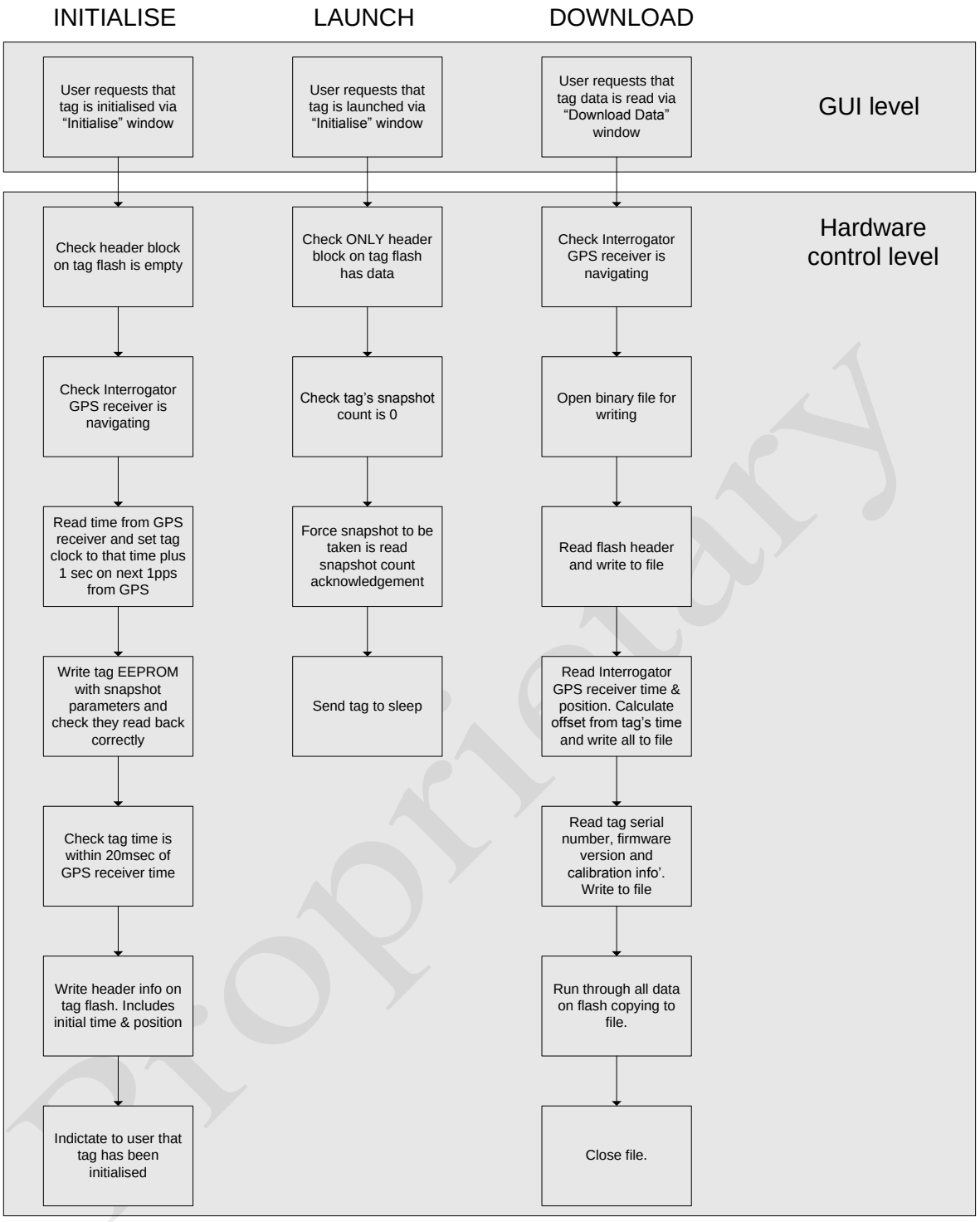


Figure 132. Flowchart for usual TrackTag operation

B.1.3 Summary

The development of the Interrogator was vital to the research strategy. This strategy involved use of customer data taken around the world in order to ensure the DSP

development was targeted in areas that meet those environment requirements. All research therefore depended on the Interrogator hardware and software being available very early on and deployed with selected customers.

The Interrogator design, in terms of hardware was reasonably straight-forward, however, the software development was a major undertaking accounting for many months of effort and is, indeed, still under development within NAVSYS. The Interrogator code totalled over 7500 lines and that's not including the GUI dialog box designs and header files.

B.2 Tag Design

Annotated photos, Figure 133 and Figure 134, show both sides of the tag. On such a small board special care had to be taken within ground planes and screening. There are a number of sub-circuits that have been split into sections on the board separated by RF screening. There are also metal screening cans designed to be soldered into place on the bare copper rectangles that can just be seen in the photos.

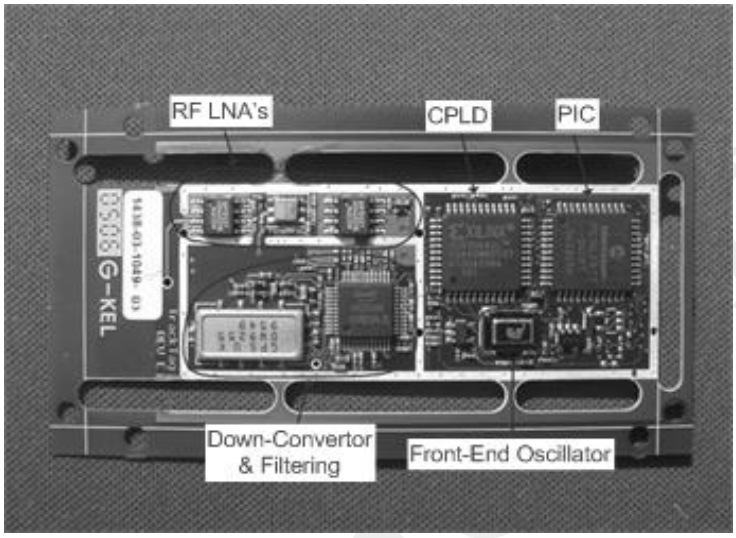


Figure 133. Tag Photo (Top Side)

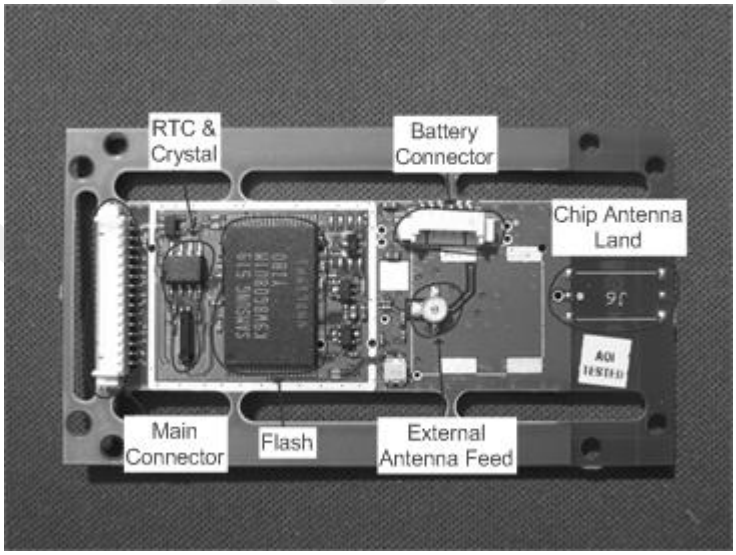


Figure 134. Tag Photo (Bottom Side)

B.2.1 Front-End Description

The front-end design is largely re-used from existing NAVSYS products. An understanding of its design is required before the development of the DSP algorithm research can be discussed. Figure 135 shows the basic building blocks that make up the RF section of the tag. Many of these components can be seen in the photos given in Figure 133 and Figure 134.

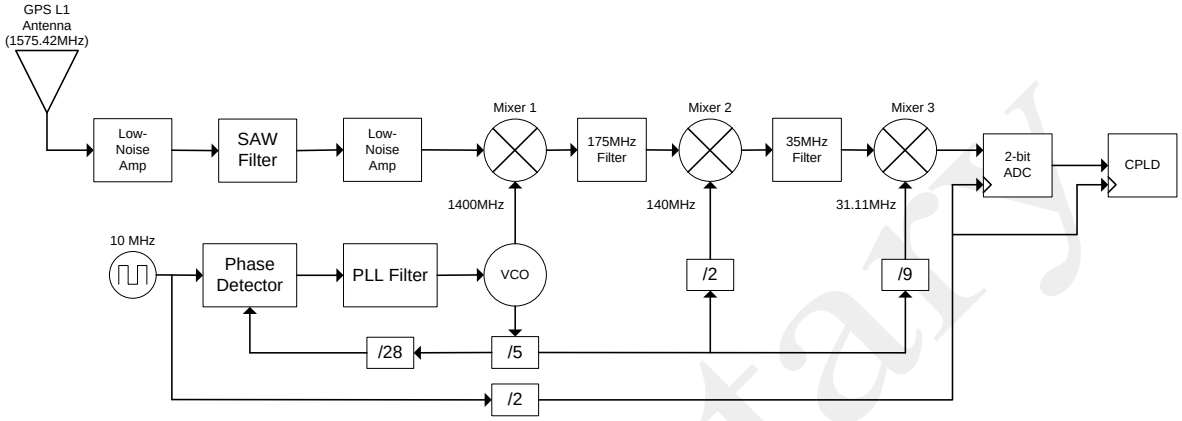


Figure 135. Front-End Block Diagram

There are three mixer stages that down-convert the L1 signal from 1575.42MHz to 4.3MHz. The same reference clock is used to provide a 5MHz sample clock. The way in which the frequency is down-converted and sampled can be seen in Figure 136.

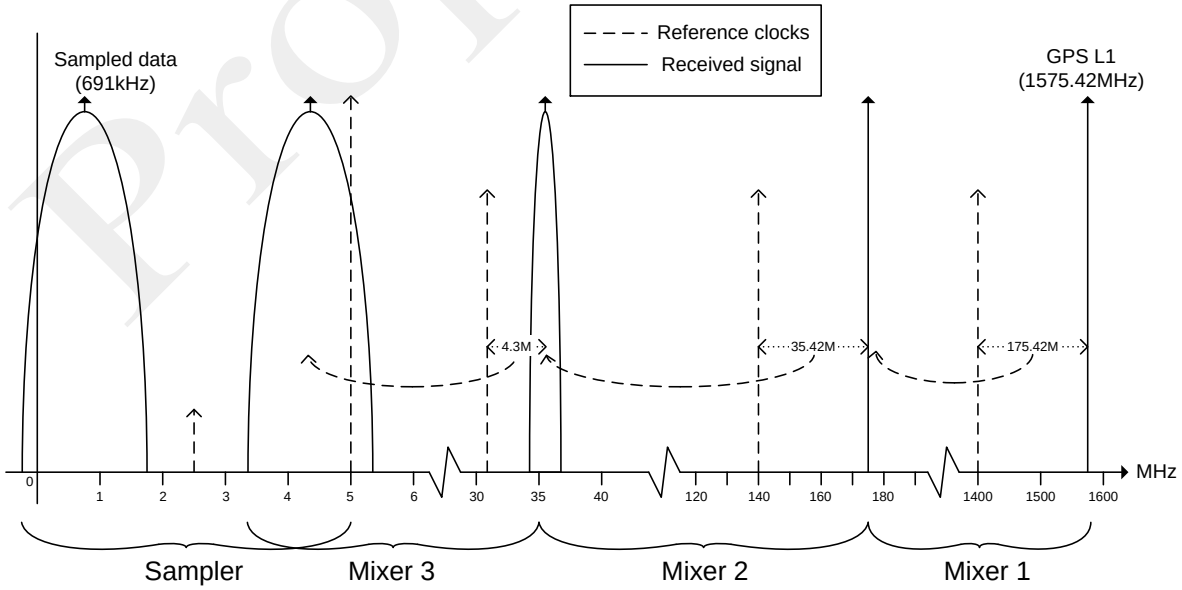


Figure 136. Front-End Frequency Plan

It was noted that under-sampling the 4.3MHz signal at 5MHz folds the signal over to 691kHz and with its 2.048MHz bandwidth some of the signal is lost. The percentage of the signal lost is actually very small, especially when considering the signal is not uniformly spread over its bandwidth. There are strong reasons behind using the 5MHz sample rate – mainly because it has to have a common reference as the down-converter to allow coherent integration on the DSP side. It is, however, non-optimal and should be considered if/when NAVSYS decide to re-design the tag's front-end.

The down-conversion process from L1 to 4.3089MHz is affected by Doppler and local oscillator drift. The nominal frequency of the intermediate frequencies will vary depending on the accuracy of the on-board 10MHz oscillator. Equation (32) describes how the oscillator's error (Δ_{osc}) affects the IF frequency.

$$\begin{aligned} IF_{freq} &= f_{L1} - 140(10^6 + \Delta_{osc}) - 14(10^6 + \Delta_{osc}) - \frac{28}{9}(10^6 + \Delta_{osc}) \\ IF_{freq} &= f_{L1} - 157.111(10^6 + \Delta_{osc}) \\ IF_{freq} &= 4.3089 \times 10^6 - 157.111\Delta_{osc} \end{aligned}$$

(32)

The actual Doppler frequency on the signal of interest will be passed through the 3 mixers and so the equation for observed frequency at the sampler is shown as equation (33);

$$IF_{freq} = 4.3089 \times 10^6 + f_{Dopp} - 157.111\Delta_{osc}$$

(33)

B.3 Algorithm Development

All algorithm development was undertaken using Matlab. There were 4 major constituent parts to the Matlab code generated for the research;

1. Signal Detection
2. Time Estimation
3. Position Estimation
4. Output Plot Generation

B.3.1 Signal Detection

The sampled signal, as it is presented to the algorithm code, requires further down-conversion to get it down to baseband. This is done using a C function embedded in the Matlab code for speed.

The shift registers required for "gold-code" generation were implemented in Matlab and sequences for all possible satellites were calculated and stored as a file. This was considered the most practical way as the codes for CA codes never change and therefore calculating them should be a one-off process.

So, the CA codes for every SV are accessed from file. These are stored as the pure 1023-bit long gold codes. To convert them to the 5000 sample values expected, a C function was written that takes a 1023 element CA code vector and assumes they are exactly spaced over 1ms. The CA code vector therefore effectively sets amplitude and time for the BPSK sequence over the entire millisecond.

A 5000 element time vector that relates the expected sample times is also passed in. The two vectors have the same time reference and therefore values for each sample are found by taking the amplitude of the CA code vector at each sample time. These values require interpolation as the sample point rarely fall on a CA chip transition. Figure 137 shows the relationship between the 2 vectors. The CA code bit width is fixed at its nominal value, i.e. 1/1.023MHz, around 200ns. The delta sample time (dt_s), however, has to be calculated based on the Doppler.

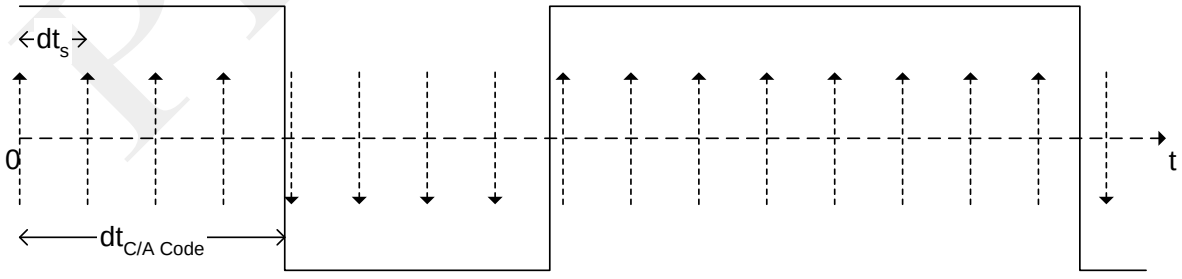


Figure 137. Gold Code Spreading

There is an added complication in that with negative Doppler the expected sample values will exceed 1ms and more than 1023 gold code bits are required. The function

handles that as well as the possibility that the CA code period of interest may be many periods, and/or fractional periods, away from the zero second time reference point.

The nominal sample period is offset using the ratio between the effective Doppler (that is the actual SV Doppler plus the apparent Doppler due to the tag oscillator offset) and the nominal carrier frequency, as shown in equation (34);

$$dt_s = \left[1 + \frac{f_{SV\_Dopp} + f_{osc\_bias}}{f_{L1}}\right] \times dt_{nom} \quad \textbf{(34)}$$

For example, if the there zero Doppler, the sample period would be equal to the nominal sample period. However, if there is a Doppler of, say, +1575.42kHz (albeit excessive and would never happen in the real system), the sample period would have 1575.42k/1575.42M = 1/1000th of a sample period added. This increase in the sample period results in some points being taken from an additional bit over and above the 1023 bits specified by the gold-code. Effectively squeezing more chips into the millisecond, this is expected when you have a higher observed frequency. Inversely, if there is a negative Doppler shift, there will be less than the full 1023 chips used and it will have the appearance of the code being stretched.

Initially, in order to search for the carrier, each bin was set to be 250Hz apart and the range was +/-4kHz. The bins searched were therefore -4000, -3750, -3500,… +3750, +4000Hz. This equated to 33 Doppler bins. Once a signal was detected which had a power greater than a certain level, the Doppler search range was reduced to +/-2kHz and therefore any subsequent satellite signals (only for that snapshot) would have 17 Doppler bins to process.

So, in the worst-case scenario, where the sky is obscured for some reason and no signal will be detected, all possible satellites will be searched for. The total number of Doppler bins to search depended on how many satellites in the constellation are expected to be within view at the given location and time. Based upon the author's experience, this number tends to be around 10. The number of bins to search in a single snapshot could be as high as 330 and that is before any attempt is made to detect weak

signals where even more Doppler bins are required due to the narrowing bandwidth when integrating over longer periods.

As the correlation function was implemented the factor of most interest was computational speed. One SV was processed for one Doppler bin and it was timed at 4.54 seconds. This suggests that processing a typical 32kB snapshot would take between 14 and 25 minutes! At that rate a tag with its 1GB Flash full of data could take 1.5 years to process and that's just for one pass of a very crude detection algorithm.

How the signal detection process was improved both in terms of optimisation and in terms of improving the performance (required for weak signal detection) is discussed in full within the Signal Detection chapters.

B.3.2 Position Estimation Algorithm

The "Position Estimation" element was fundamental to the work though as this was the sub-system that took the inputs from the research activities to generate output data for comparison. It was therefore necessary to keep the position estimation code simple and stable so that comparisons over the course of the research work would be fair and consistent. The "Output Plot Generation" element was necessary to support the work being done by biologists in the field gathering data for TrackTag's development as well as their own interests. Figure 138 shows the top level design implemented for the navigation solution.

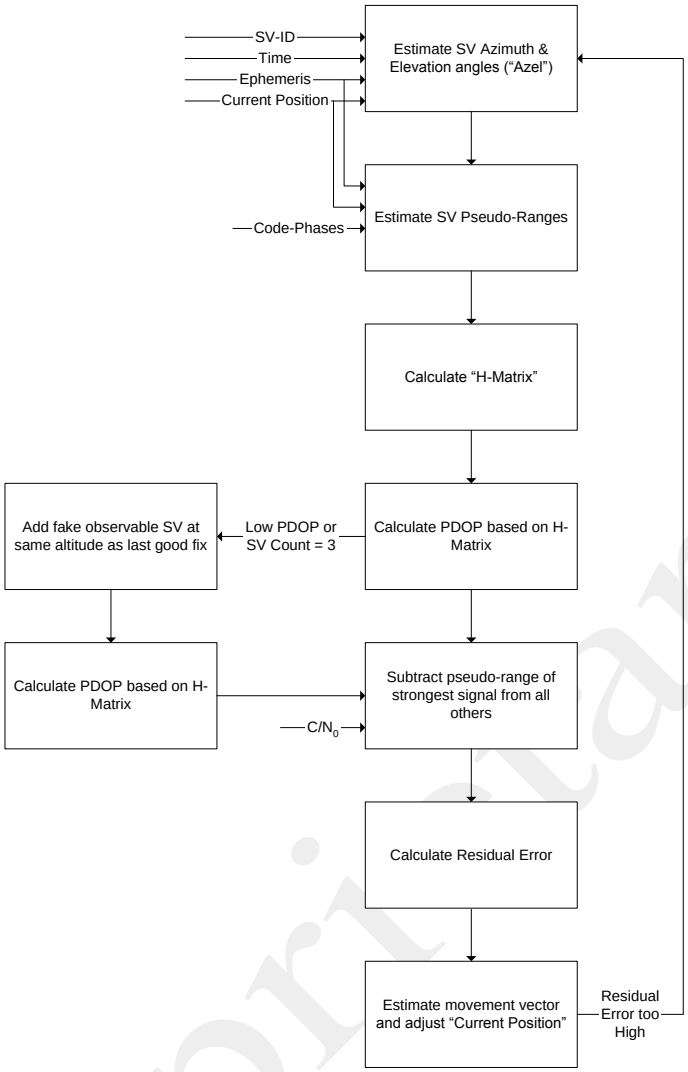


Figure 138. Navigation Solution Top Level

The navigation code itself follows very basic routine for estimation of the position based on pseudo-ranges (as discussed in the section on “GPS Fundamentals).

Appendix C: SBAS Broadcast Data File Description

Consolidated SBAS messages archived and available from ftp://serenad-public.cnes.fr/SERENAD/FROM_CNES/MSG/year.

File naming convention is M*prnddd*0.*yy*b where;

prn:- Satellite PRN

ddd:- Julian Day of year

yy:- Year

ASCII Text file description is as follows;

TABLE S-1

GEOSTATIONARY BROADCAST DATA FILE - HEADER SECTION DESCRIPTION

HEADER LABEL (Columns 61-80)	DESCRIPTION	FORMAT
RINEX VERSION / TYPE	- Format version (2.10)	F9.2,11X,
	- File type ('B' = SBAS binary broadcast messages)	A1,39X
PGM / RUN BY / DATE	- Name of program creating current file	A20,
	- Name of agency creating current file	A20,
	- Date of file creation (dd-mmm-yy hhmm)	A20
COMMENT	Comment line(s) *	A60
REC INDEX/TYPE/VERS	- Receiver index. File-internal index * used by "PRN / EPOCH / RCVR" record to reference receiver collecting the data(0<index<=99)	3X,I3,14X,
	- Receiver type	A20,
	- Rcvr version (e.g., Firmware version) Record repeated for every receiver type	A20
END OF HEADER	Last record in the header section.	60X

Records marked with * are optional

TABLE S-2

GEOSTATIONARY BROADCAST DATA FILE - DATA RECORD DESCRIPTION

OBS. RECORD	DESCRIPTION	FORMAT
PRN / EPOCH / RCVR	- Identifier number: SBAS: [PRN] (currently defined range: 120-158) Services w/o satellite transmitters can use the identifier to distinguish between different servers involved. Use COMMENT lines to further describe the servers.	I3,1X

	- Epoch of data block (GPS time of reception of start of data block) [year (2 digits, padded with 0), month, day, hour, minute, second]	I2.2, 4*(1X,I2),F5.1, 2X
	- Band (Currently L1 only)	A2,3X
	- Length of data message (bytes)	I3, 3X
	- SBAS:the message currently consists of 32 bytes (original 250 bits of the transmitted data block plus 6 trailing zero-bits) and optional receiver-generated CRC parity bytes.	I3, 3X,
	- File-internal receiver index(see REC INDEX/TYPE/VERS record, 0 if not applicable)	
	- Transmission System Identifier. SBA : data block transmitted by SBAS satellite SNT : data block transmitted by SiSNET CDG : data block transmitted by CDGPS 000 : not known	A3
DATA RECORD - 1	-Message frame identifier ex : SBAS Message type derived from the 6 MSBs of the second data byte.	1X,I2,4X,
	- First 18 bytes of message (hexadecimal)	18(1X,Z2.2)
DATA RECORD - 2	Remaining bytes of message (hex). For messages longer than 36 bytes the DATA RECORD - 2 is repeated as often as necessary.	7X, 18(1X,Z2.2)

References

1. Acharya, L.F., M. B., *Echolocation behaviour of vespertilionid bats (lasiurus cinereus and lasiurus borealis) attacking airborne targets including arctiid moths.* Canadian Journal of Zoology, 1992. **70**: p. 1292–1298.
2. Avery, M.A.S.P.F.C.J.F., *The effects of a tall tower on nocturnal bird migration - a portable ceilometer study.* The Auk, 1976. **93**(2): p. 281-291.
3. Griffin, D.R., *Airplane observations of homing pigeons.* Bull. Mus. Comp. Zool., 1952. **107**: p. 411–440.
4. Le Munyan, C.D., W. White, E. Nybert, and J. J. Christian, *Design of a miniature radio transmitter for use in animal studies.* Journal of Wildlife Management, 1959. **23**: p. 107-110.
5. E., E., *A method for measuring the heart rate and stroke/pulse pressures of birds in normal flight.* Arbok Universitet Bergen, Matematisk Naturvitenskapelig, 1960. **12**: p. 1-22.
6. Michener, M.C. and C. Walcott, *Navigation of single homing pigeons: Airplane observations by radio tracking.* Science, 1966. **154**: p. 410–413.
7. MacLean, G., *TrackTag Marketing Strategy Document.* 2004, Institute for System Level Integration: Livingston.
8. Brander, R.B. and W.W. Cochran, *Radio Location Telemetry.* Wildlife Management Techniques, 1969: p. 95-103.
9. M. A. Fedak, P.L.B.J.M., *MAMVIS: A Marine Mammal Behaviour Visualization System.* Journal of Visualization and Computer Animation, 1996. **7**: p. 141-147.
10. Nussberger, B.I., *Effects of radio-collars on behaviour of alpine chamois Rupicapra rupicapra rupicapra.* Wildlife Biology, 2006. **12**: p. 339-343.
11. M.C., G., *Effects of radiotagging on the weight gain and survival of Curlew Numenius arquata chicks.* Bird Study, 2002. **49**: p. 172-176.
12. BJ McConnell, R.B., E Bryant, C Hunter, P Lovell, AJ Hall, *Phoning home - a new GSM mobile phone telemetry system to collect mark-recapture data.* Marine Mammal Science, 2004. **20**: p. 274-283.
13. White, G.C., *Optimal locations of towers for triangulation studies using biotelemetry triangulation data.* Journal of Wildlife Management, 1985. **49**: p. 190-196.
14. L. Scott Mills, F.F.K., *Observer Performance in Known and Blind Radio-Telemetry Accuracy Tests.* The Journal of Wildlife Management, 1989. **53**(2): p. 340-342.
15. Garrott, R.A.W., G. C. & Bartmann, R., *Reflected signal bias in biotelemetry triangulation systems.* Journal of Wildlife Management, 1986. **50**(4): p. 747-752.
16. Lee, J.E., White, G. C, Garrott R. A., Bartmann, R. M. & Alldredge, A. W., *Assessing the accuracy of a radiotelemetry system for estimating animal locations.* Journal of Wildlife Management, 1985. **49**(3): p. 658-663.
17. Misra, P. and P. Enge, *Global Positioning System. Signals, Measurements, and Performance*. 2001, Lincoln, Massachusetts: Ganga-Jamuna Press. 390.
18. Hutchinson, J.F.K.a.C.E., *The OMEGA Navigation System - An Overview.* IEEE Communications Society Magazine, 1978: p. 23-35.
19. Sakran, F.C., Jr. Swanson, E.R. . *Omega: the end-what it was, what it did, what now?* in *Position Location and Navigation Symposium* 1998: IEEE.

20. J. Bishop & J. D. Last. *An Omega radio-navigation receiver for animal tracking, University of Wales, Bangor, UK*. in *Radio Receivers and Associated Systems*. 1995: IEE.
21. GmbH, O.-S., *SAFIR, SAtellite For Information Relay, Users Guide*. 1993, OHB-System GmbH.: Bremen, Germany.
22. W. H. Guier, G.C.W., *Genesis of Satellite Navigation.* John Hopkins APL Technical Digest, 1997. **18(2)**: p. 178-181.
23. Danchik, R.J., *An Overview of Transit Development.* John Hopkins APL Technical Digest, 1998. **19(1)**.
24. *Basic Description of the Argos System*. [cited; Available from: http://www.argosinc.com/system_overview.htm.
25. Clark, D.D. *Overview of the Argos System*. in *OCEANS '89*. 1989.
26. Deutsch, C., R.K. Bonde, and J.P. Reid, *Radio–tracking manatees from land and space: Tag design, implementation, and lessons learned from long–term study.* Mar. Technol Soc. J., 1998. **32**: p. 18–29.
27. Hatase, H., et al., *Pelagic habitat use by an adult Japanese male loggerhead turtle Caretta caretta examined by the ARGOS satellite system.* Fish. Sci., 2002. **68**: p. 945–947.
28. Jay, C.V. and G.W. Garner, *Performance of a satellite–linked GPS on Pacific walruses (Odobenus rosmarus divergens).* Polar Biol., 2002. **25**: p. 235–237.
29. Stokesbury, M., et al., *Movement and environmental preferences of Greenland sharks (Somniosus microcephalus) electronically tagged in the St. Lawrence Estuary, Canada.* Mar. Biol., 2005. **148**(1): p. 159-165.
30. Pennycuick, C.J., et al., *Response to weather and light conditions of migrating Whooper Swans Cygnus cygnus and flying height profiles, observed with the Argos satellite system.* Ibis, 1999. **141**: p. 434–443.
31. Biro, D.M.J.G.T., *Route recapitulation and route loyalty in homing pigeons:pilotage from 25 km?*, in *Proc. Conf. RIN 05*. 2005, R. Inst. Nav.: Reading University, UK.
32. Biro, D., et al., *From Compromise to Leadership in Pigeon Homing.* Curr. Biol., 2006. **16**: p. 2123-2128.
33. Biro, D., et al., *How the viewing of familiar landscapes prior to release allows pigeons to home faster, evidence from GPS tracking.* J. Exp. Biol., 2002. **205**: p. 3833–3844.
34. Dennis, T.E., M.J. Rayner, and M.M. Walker, *Evidence that pigeons orient to geomagnetic intensity during homing.* Proc. R. Soc. B., 2007. **FirstCite Early Online Publishing**.
35. von Hünerbein, K., E. Rüter, and W. Wiltschko, *Flight Tracks of Homing Pigeons Measured with GPS.* 2001.
36. Hulbert, I.A.R. and J. French, *The accuracy of GPS for wildlife telemetry and habitat mapping.* J. Appl. Ecol., 2001. **38**: p. 869–878.
37. Girard, I., et al., *Effects of sampling effort based on GPS telemetry on home–range size estimations.* J. Wildl. Manage., 2002. **66**: p. 1290–1300.
38. Meyburg, B.U., et al., *GPS Satellite Tracking of Lesser Spotted Eagles Aquila pomarina: home range and territorial behaviour in the breeding area.* Vogelwelt, 2006. **127**: p. 127-144.
39. Hawkins, P., *Refinements in telemetry procedures: applications for studies of animal navigation*, in *Proc. Conf. RIN 05, Orientation and Navigation - Birds, Humans and Other Animals*. 2005, R. Inst. Nav.: Reading University, UK.

40. Vyssotski, A.L., et al., *Miniature Neurologgers for Flying Pigeons: Multichannel EEG and Action and Field Potentials in Combination With GPS Recording.* J. Neurophysiol., 2006. **95**: p. 1263-1273.
41. Lindzey, M.A.R.F., *Effects of forest vegetation and topography on Global Positioning System collars for elk.* ACSM/ASPRS, 1997: p. 492-501.
42. Haddrell, D.T.M.T., *GPS in Autonomous Underwater Vehicles.* Electronic Engineering in Oceanography, 1994.
43. Moernaut, D.O.G.J.K., *The Basics of Patch Antennas.* Globalnet's RF & Microwave Solutions Update, 2006.
44. Wingfield, O.P.L.A. *Miniature dielectrically loaded GPS antennas for robust US E-911 position location*. in *Personal Mobile Communications Conference*. 2003: IEEE.
45. O. P. Leisten, J.C.V., P. McEvoy, R. Seager & A. Wingfield, *Miniature dielectrically-loaded quadrifilar antenna for Global Positioning System (GPS).* Electronic Letters, 2001. **37**: p. 1321.
46. *Interface Control Document (ICD) GPS (200)*, N.G.S. Segment, Editor. 1992.
47. Peterson R L, Z.R.E., Borth D E, *Introduction to Spread-Spectrum Communications*. 1995: Prentice Hall.
48. Parkinson, B.W. and J.J. Spilker Jr, *Global Positioning System: Theory and Applications.* Progress in Astronautics and Aeronautics, 1996. **163**.
49. *SiRFstarIII GSC3e/LP Datasheet*. 2007 [cited; Available from: http://www.sirf.com/products/gps_chip3e.html.
50. M. D. Karunanayake, M.E.C.G.L., *Analysis of assistance data on AGPS performance.* Measurement Sci. Tech., 2007. **18**: p. 1908-1916.
51. Bryant, E. *Fastloc Technology*. [cited; Available from: www.wildtracker.com/fastloc.
52. Brown, A. *The TIDGET - A Low Cost GPS Sensor for Tracking Applications.* in *ION Satellite Division International Technical Meeting*. Sept. 1992. Albuquerque, NM.
53. A. Matini, A.B.a.J.L. *Wind Profile Estimation Using a TIDGET® Payload on Weather Balloons*. in *Institute Of Navigation's 52nd Annual Meeting*. June 1996. Cambridge, MA.
54. J. Coetsee, A.M.a.A.B. *A TIDGET® / Inertial Missile Sensor Fusion System*. in *AIAA Conference*. Aug. 1995. Baltimore, MD.
55. Brown, A.K. and M.A. Sturza, *GPS Tracking System*, U.S. Patent, Editor. 2003: US.
56. *SiRF Company Technology Website*. 2005 [cited; Available from: http://www.sirf.com.
57. *Microwave Telemetry Company Website*. 2005 [cited; Available from: http://microwavetelemetry.com.
58. Schoch, D.L.a.M., M A, *Feasibility of a virtual anechoic chamber: the ultimate E3 test facility*, in *Southcon/94*. 1994: Orlando, FL.
59. G. Lachapelle, M.G.P., *Weak Signal Tracking & High Sensitivity GPS for Indoors*. 2004, Navtech Seminars & GPS Supply, Inc: ION GNSS 2004.
60. M. K. Simon, J.K.O., R. A. Scholtz and B. K. Levitt, *Spread Spectrum Communications*. Vol. 1-3. 1985: Computer Science Press.
61. L., P.M. *Block Acquisition of Weak GPS Signals in a Software Receiver*. in *ION GPS 2001*. 2001. Salt Lake City, UT.
62. C Strassle, D.M., H Mathis & C Burgi. *The Squaring-Loss Paradox*. in *GNSS 20th International Technical Meeting of the Satellite Division*. 2007. Forth Worth, TX.

63. J. A. A. Rodriguez, V.H., T. Pany & B. Eissfeller. *Theory on Acquisition Algorithms for Indoor Positioning*. in *12th Saint Petersburg International Conference on integrated navigation systems*. 2005. Saint Petersburg, Russia.
64. Dedes, G.D., A.G. . *Indoor GPS positioning - challenges and opportunities*. in *Vehicular Technology Conference*. 2005 IEEE.
65. BZheng, B.L., G. *GPS Software Receiver Enhancements for Indoor Use*. in *ION GNSS 18th International Technical Meeting of the Satellite Division*. 2005. Long Beach, CA: ION.
66. Psiaki, M.L. *Block Acquisition of Weak GPS Signals in a Software Receiver*. in *ION GPS 2001*. 2001. Salt Lake City, UT.
67. Han, C.Y.S. *Block-Accumulating Coherent Integration over Extended Interval (BACIX) For Weak GPS Signal Acquisition*. in *GNSS 19th International Technical Meeting of the Satellite Division*. 2006. Fort Worth, TX.
68. Chan-Woo Park, S.C., Jeongbok Yang, Jaeseung Yoon, Yonserk Kim, *High Sensitivity Assisted GPS Receiver Using Self-Generated SA Data for PNS Enabled Mobile Phones*, in *ION GNSS 18th International Technical Meeting of the Satellite Division*. 2005: Long Beach, CA.
69. Cutnell, J.D.a.J., K. W., *Physics*. 2nd ed. 1992: John Wiley & Sons Inc.
70. Gurtner, W. *RINEX The Receiver Independent Exchange Format Version 2.10*. [cited; Available from: ftp//igscb.jpl.nasa.gov/igscb/data/format/rinex210.txt.
71. *FAA-E-2892B Wide Area Augmentation System (WAAS) Specification*, F.A. Administration, Editor. 1999.
72. G. Schofield, C.B., G. MacLean, P. Brown, M. Baker, K. A. Katselidis, P. Dimopoulos, J. D. Pantis & G. C. Hay, *Novel GPS tracking of sea turtles as a tool for conservation management.* The Journal of Experimental Marine Biology and Ecology, 2007.
73. G Maral, M.B., *Satellite Communications Systems*. 3rd ed. 1998: Wiley.
74. Phani K. Sagiraju, D.A., David Magee, *A Fast Acquisition Algorithm for Indoor Software Receivers*, in *Institute of Navigation Annual Meeting*. 2005: Cambridge, MA.
75. Wayne Ballantyne, G.T., Gary Slimak, John Shewfelt, *Achieving Low Energy-per-fix with A-GPS Cellular Phones*, in *Institute of Navigation GNSS Technical Meeting of the Satellite Division*. 2005: Long Beach, CA.
76. Alison Brown, N.G., Keith Taylor. *Modeling and Simulation of GPS Using Software Signal Generation and Digital Signal Reconstruction*. in *Institute of Navigation National Technical Meeting*. 2000. Anaheim, CA.
77. Jose Angel Avila Rodriguez, V.H., Thomas Pany, Bernd Eissfeller, *Theory on Aquisition Algorithms for Indoor Positioning*, in *12th Saint Petersburg International Conference on integrated navigation systems*. 2005: Saint Petersburg, Russia.
78. Frank van Diggelen, C.A., *Indoor GPS Technology*, in *CTIA Wireless-Agenda*. 2001: Dallas, TX.
79. Marzluff, J.J.M.a.J.M., *Radio Tracking and Animal Populations*. 2001: Academic Press.
80. Richharia, M., *Satellite Communication Systems*. 2nd ed. 1999: MacMillan.
81. Parks, C.S.B.a.T.W., *DFT/FFT and Convolution Algorithms*. 1985, New York: John Wiley & Sons.
82. Dixon, R.C., *Spread Spectrum Systems with Commercial Applications*. 3rd ed. 1994, New York: John Wiley & Sons.
83. Peterson, R.E.Z.a.R.L., *Digital Communications and Spread Spectrum Systems*. 1985, New York: MacMillan Publishing Company.

84. Blahut, R.E., *Fast Algorithms for Digital Signal Processing*. 1984: Addison-Wesley.
85. Cochran, W.W., *Wildlife Telemetry*. Wildlife Management Techniques Manual, 1980: p. 507-520.